NGA.STND.0062_1.0_WFS

2017-01-11

NGA STANDARDIZATION DOCUMENT

National System for Geospatial-Intelligence (NSG) Web Feature Service 2.0 Implementation Profile

(2017-01-11)

Version 1.0.0

NATIONAL CENTER FOR GEOSPATIAL INTELLIGENCE STANDARDS

Table of Contents

List of Figures

List of Tables

Executive Summary

This document defines the NSG Profile of the DGIWG – Web Feature Service 2.0 Profile of ISO 19142:2010 - Web Feature Service (WFS).

The Web Feature Service provides access to geospatial features in a manner independent of the underlying data store. WFS can also provide the capability to perform operations to create, update, and delete features from a data store.

The intention of this Profile of the WFS is to promote interoperability between elements of the U.S. Intelligence Community (IC) and Department of Defense (DoD), both within and outside of the National System of Geospatial-Intelligence (NSG). Like most profiles, this specification defines the minimum set of service requirements necessary to ensure usability in the NSG. However, it goes beyond that scope to define the extensions and restrictions necessary to support the following:

- Integration with other IC and DoD standards for Discovery and Retrieval.
- Integration with the IC and DoD Attribute Based Access Control (ABAC) infrastructure.

Sources for the capabilities supported by this Profile include:

- NGA Consolidated Libraries (NCL) Program Management Office (PMO) and developers
- ODNI Content Discovery and Retrieval Standards
- ES&IS Program
- NGA Content Portfolio
- NGA GLOBE Initiative
- U.S. Army Geospatial Center (AGC)

i. Document contributor contact points

Person	Organization	Email
Charles Heazel	WiSC Enterprises	cheazel@wiscenterprises.com

ii. Future work

Version 1.0.1

Implementation and conformance testing resources cannot be constructed prior to approval of the specification which they support. Such resources include XML Schema, Compliance Tests, Reference Implementations, and the Implementation Conformance Statement. Work on these resources will commence once Version 1.0.0 of this Profile has been finalized. That work will result in modifications to sections 4.3.3 and 4.3.4. A template for the Implementation Conformance Statement will also be added as Annex D. These changes are all informative and will have no impact on existing implementations.

OGC Related

This Profile of WFS 2.0 was developed in parallel with Open Geospatial Consortium (OGC) efforts that could duplicate capabilities documented in this Profile. Ideally, NSG requirements would be addressed in an OGC standard rather than in a Profile. Therefore, as these efforts progressed, capabilities originally planned for this Profile have been delegated to the appropriate OGC effort.

Testbed 12

The OGC conducts annual rapid prototyping initiatives call Testbeds. The purpose of these Testbeds is to explore new technology and its application to geospatial interoperability. The following activities from this years Testbed (Testbed 12) are applicable to this Profile.

- **REST**

  OGC Testbed 12 has produced two Engineering Reports related to RESTful Web Services.

  - REST Architecture Engineering Report (OGC 16-035)
  - REST Users Guide (OGC 16-057r1)

  These documents seek to better define the characteristics of a RESTful service and to establish how RESTful and SOA services can interoperate in a distributed enterprise. This work will inform the OGC WFS 2.5 efforts.

- **JSON**

  OGC Testbed 12 investigated what is involved in adding JSON-based formats to the encodings supported by OGC services. Two Engineering Reports were produced:

  - JavaScript-JSON-JSON-LD Engineering Report (OGC 16-051)
  - TopoJSON Engineering Report (OGC 16-056)

These reports provide the technical foundation for handling OGC Feature data using JSON, GeoJSON, TopoJSON, JSON Schema, and JSON-LD. This work will inform the OGC WFS 2.5 efforts.

- **Asynchronous Services**

Advanced analytic models such as ABI require different model for discovery and access. An analyst does not issue a query to discover a resource. Rather, they issue a request that the service inform them whenever a suitable resource is available. This means that the response to a request may come hours, days, or even weeks after the request. OGC Testbed 12 investigated an asynchronous version of the WFS which could support this model. That work is captured in the following Engineering Report:

- Implementing Asynchronous Service Response Engineering Report (OGC 16-023r2)

This work is planned for inclusion in the OGC WFS 2.5 and Filter Encoding 2.5 standards.

- **Authentication Advertisement**

Testbed 12 developed and validated a means to advertise the Identification and Authentication methods supported by a service. That work is documented in:

- OWS Common Security Engineering Report (OGC 16-048r1)

This work will be incorporated into the OGC efforts to support secure services.

Secure Services

The OGC is working to extend the OWS Common (OGC 16-121r9) standard to better support information assurance requirements. OGC Testbed 12 made a start by providing a means to advertise the Identification and Authentication methods supported by a service. This work will continue through the OWS Common Security Standards Working Group and Testbed 13.

GWG Web Services Focus Group members are participating in the OWS Common Security Standards Working Group. Their goal is to address additional Information Assurance requirements. These requirements include better support for HTTPS, Secure REST services, and identity delegation though a service chain. While it is not clear if or how this work will fold into OGC standards, it will serve as an OGC endorsed capability for use in the NSG profiles.

OGC WFS 2.5 and Filter Encoding 2.5

The OGC is working on an update to the Web Feature Service and its associated Filter Encoding specifications. It is anticipated that most of the Testbed 12 work described above will be incorporated into the new standards. GWG Web Services Focus Group members are participating in the WFS-FES Standards Working Group.

NSG Profile of WFS 2.5

Following release of the OGC WFS 2.5 and Feature Encoding 2.5 standards, work will begin on an NSG Profile of WFS 2.5. Prior to committing on a new profile, the GWS Web Services Focus Group will assess how well the new OGC standards address NSG needs and identify the best way to address any gaps. Areas of investigation will include:

- Support for advanced analytics

- Disrupted, Disconnected, Intermittent and Limited bandwidth (DDIL) environments
- Security enhancements
- SOM and OBP support

1 Introduction

This document defines the NSG Profile of the DGIWG – Web Feature Service 2.0 Profile of ISO 19142:2010 - Web Feature Service (WFS).

The Web Feature Service provides access to geospatial features in a manner independent of the underlying data store. WFS can also provide the capability to perform operations to create, update, and delete features from a data store.

The intention of this Profile of the WFS is to promote interoperability between elements of the U.S. Intelligence Community (IC), Department of Defense (DoD), NATO, and coalition partners. While inherently a GEOINT service, this Profile includes extensions to better integrate with services and users both within and outside of the National System of Geospatial-Intelligence (NSG). Those extensions and restrictions are designed to achieve the following:

- Maintain compatibility with OGC and DGIWG versions of this standard.
- Integration with other IC and DoD standards for Discovery and Retrieval.
- Support for the IC and DoD Attribute Based Access Control (ABAC) infrastructure.
- Support for time-based versioning of content

2 References

2.1 Normative References

The following normative documents contain provisions, which, through reference in this text, constitute provisions of this document. For dated references, subsequence amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

1. DGIWG 122, *DGIWG – Web Feature Service 2.0 Profile*, (November 16, 2015)
2. IETF RFC 4122, *A Universally Unique IDentifier (UUID) URN Namespace*, (July 2005)
3. ISO/IEC 9075-1:2011 *Part 1: Framework (SQL/Framework)*
4. ISO/IEC 9075-1:2011 *Part 2: Foundation (SQL/Foundation)*
5. ISO 19136:2007, *Geographic information — Geography Markup Language (GML)* (1 September 2007)
6. JSR-338, *Java™ Persistence API, Version 2.1* (02 April 2013)
7. ODNI, *Text and XML Data Encoding Specifications for Intelligence Community Identifier*, Version 1 (10 April 2013)
8. ODNI, *XML Data Encoding Specification for Information Security Markings*, Version 13 (9 May 2014)
9. OGC 06-121r3, *OGC Web Services Common Specification, OGC® Implementation Standard* (9 February 2009)
10. OGC 07-092r3, *Definition identifier URNs in OGC namespace, OGC® Best Practices* (15 January 2009)
11. OGC 09-025r2, *Web Feature Service 2.0 Interface Standard – With Corrigendum.* (10 July 2014)

12. OGC 09-026r2, *Filter Encoding 2.0 Encoding Standard.* (18 August 2014)
13. W3C SOAP, *Simple Object Access Protocol (SOAP) 1.2,* W3C Note (27 April 2007)
14. W3C SPARQL, *SPARQL 1.1 Query Language*, W3C Recommendation (21 March 2013)
15. W3C XPath, *XML Path Language (XPath) Version 1.0*, W3C Recommendation (16 November 1999)
16. W3C XQuery, *XQuery 1.0: An XML Query Language*, W3C Recommendation (08 April 2014)

2.2 Informative References

The following informative documents provide context for this Profile. Although many of these documents are referenced in this standard, they are not normative for any implementations.

1. IETF RFC 3986, *Uniform Resource Identifier (URI): Generic Syntax*, January 2005
2. ISO/TS 19103:2005, *Geographic information — Conceptual schema language,*
3. ITU-T X.509, *The Directory: Public-key and attribute certificate frameworks*, (October 2012)
4. NGA.RP.0001_1.0.0, *National System for Geospatial-Intelligence Recommended Practice for Universally Unique Identifiers*, (3 January 2013)
5. NSG Directive 3201, *The Geospatial Intelligence (GEOINT) Functional Manager Standards Assessment (GFMSA) Program*, (June 2015)
6. NSG Manual 3202, *GEOINT Functional Manager Standards Assessment Program*, (9 June 2016)
7. ODNI, *Atom Data Encoding Specification for CDR Result Sets*, Version 2.0 (10 April 2013)
8. ODNI, *Keyword Query Language*, Version 2.0 (3 October 2012)
9. ODNI, *REST Interface Encoding Specification for CDR Brokered Search*, Version 2.0 (6 September 2013)
10. ODNI, *SOAP Interface Encoding Specification for CDR Brokered Search*, Version 2.0 (6 September 2013)
11. ODNI, *REST Interface Encoding Specification for CDR Retrieve*, Version 2.0 (3 October 2012)
12. ODNI, *SOAP Interface Encoding Specification for CDR Retrieve*, Version 2.0 (3 October 2012)
13. ODNI, *REST Interface Encoding Specification for CDR Search*, Version 3.0 (3 October 2012)
14. ODNI, *SOAP Interface Encoding Specification for CDR Search*, Version 3.0 (3 October 2012)
15. ODNI, *REST Service Encoding Specification for CDR Deliver*, Version 2.0 (9 May 2014)
16. ODNI, *SOAP Service Encoding Specification for CDR Deliver*, Version 2.0 (9 May 2014)
17. ODNI, *REST Service Encoding Specification for CDR Query Management*, Version 2.0 (9 May 2014)

18. ODNI, *REST Service Encoding Specification for CDR Query Management*, Version 2.0 (9 May 2014)
19. OpenSearch.org, *OpenSearch*, Version 1.1 Draft 5, http://www.opensearch.org/Specifications/OpenSearch/1.1
20. W3C WSDL, *Web Services Description Language (WSDL) 1.1,* W3C Note (15 Mar 2001)
21. W3C XML Namespaces, *Namespaces in XML*, W3C Recommendation (14 January 1999)
22. W3C XML Schema Part 1, *XML Schema Part 1: Structures*, W3C Recommendation (2 May 2001)
23. W3C XML Schema Part 2, *XML Schema Part 2: Datatypes*, W3C Recommendation (2 May 2001)

3 Terms, Definitions, and Abbreviations

For the purposes of this document, terms, acronyms, and definitions found in WFS 2.0 [OGC 09-025r2], OGC Filter Encoding (OGC 09-026r2), and the DGIWG WFS Profile [DGIWG 122] apply.

In addition, the following terms, acronyms, and definitions shall apply:

3.1 Definitions

1. **abstract test case (GFMSA)**
   a generalized test for a particular requirement.
2. **abstract test method (GFMSA)**
   a method for testing an implementation that is independent of any particular test procedure.
3. **abstract test module (GFMSA)**
   a set of related abstract test cases. Abstract test modules may be nested in a hierarchical way.
4. **Abstract Test Suite (ATS) (GFMSA)**
   a set of abstract test modules and associated abstract test cases that collectively specify all the requirements to be satisfied for conformance.
5. **Activity-Based Intelligence**
   An Intelligence Community (IC) analytic method applied to structured data from all sources, to discover objects, relationships, or behaviours by resolving significant activity.
   [ABI 2017 Concept of Operations Version 4.0 September 2015]
6. **attribute <XML>**
   name-value pair contained in an **element** (18)
   [ISO 19136:2007, definition 4.1.3]
   NOTE In this document an attribute is an XML attribute unless otherwise specified.
7. **attribute (UML)**
   the description of a named slot of a specified type in a **class** (8).
   [The Unified Modelling Language Reference Manual, 1999]
8. **basic test (GFMSA)**
   an initial capability test intended to identify clear cases of non-conformance.
9. **capability test (GFMSA)**
   a test designed to determine whether an IUT conforms to a particular characteristic of a standard as described in the test purpose.

10. **Compliance (GFMSA)**
    adherence to policy, directives, instructions, guidance, etc. Often used to define or mean the same as conformance. E.g. an implementation exhibits conformance if it complies with the conformance requirements of the applicable information standards.
11. **Conformance (GFMSA)**
    the fulfilment of specified requirements.
12. **conformance Class (GFMSA)**
    conformance classes may be used to group, define, and label different kinds of conformance requirements pertinent to implementation of the standard.
13. **class (UML)**
    the descriptor for a set of objects that share the same attributes, operations, methods, relationships, and behaviour.
    [The Unified Modelling Language Reference Manual, 1999]
14. **client**
    software component that can invoke an **operation** (34) from a **server** (46)
    [ISO 19128:2005, definition 4.1]
15. **coordinate**
    one of a sequence of n numbers designating the position of a point in n-dimensional space
    [ISO 19111:2007, definition 4.5]
16. **coordinate reference system**
    coordinate system (17) that is related to an object by a datum
    [ISO 19111:2007, definition 4.8]
17. **coordinate system**
    set of mathematical rules for specifying how **coordinates** (15) are to be assigned to points
    [ISO 19111:2007, definition 4.10]
18. **element <XML>**
    basic information item of an XML document containing child elements, **attributes** (1) and character data
    [ISO 19136:2007, definition 4.1.23]
19. **feature**
    abstraction of real world phenomena
    [ISO 19101:2002, definition 4.11]
    NOTE A feature can occur as a type or an instance. The term "feature type" or "feature instance" should be used when only one is meant.
20. **feature identifier**
    identifier that uniquely designates a **feature** (19) instance
21. **filter expression**
    predicate expression encoded using XML
    [OGC 09-026r214-103, definition 4.11]
22. **Implementation Conformance Statement (ICS) (GFMSA)**
    a statement made by the supplier of an implementation or system claimed to conform to a given standard (or set of standards/specifications), asserting which capabilities have been conformingly implemented.
23. **Implementation Under Test (IUT) (GFMSA)**
    the realization of a specification that is the focus of test.
24. **interface**
    named set of **operations** (34) that characterize the behaviour of an entity

[ISO 19119:2005, definition 4.2]

25. **join predicate**
filter expression (21) that includes one or more clauses that constrain properties from two different entity types
[OGC 09-026r214-103, definition 4.16]
NOTE In this International Standard, the entity types will be **feature** (19) types.

26. **join tuple**
set of two or more object instances that satisfy a filter that includes **join predicates** (25)
NOTE In this International Standard, the object instances will be **feature** (19) instances.

27. **local resource**
resource that is under the direct control of a system
NOTE In this International Standard, the system is a web feature service and the resource in held in a data store that is directly controlled by that service.

28. **locator attribute**
attribute (1) whose values is a reference to a **local resource** (27) or **remote resource** (38)
NOTE: In XML, this attribute is commonly called an href and contains a URI reference to the remote resource (see W3C XLink).

29. **Multipurpose Internet Mail Extensions (MIME) type**
media type and subtype of data in the body of a message that designates the native representation (canonical form) of such data
[IETF RFC 2045:1996]

30. **namespace <XML>**
collection of names, identified by a URI reference which are used in XML documents as **element** (18) names and **attribute** (1) names
[W3C XML Namespaces:1999]

31. **object (UML)**
A discrete entity with a well-defined boundary and identity that encapsulates state and behaviour; an instance of a class.
[The Unified Modelling Language Reference Manual, 1999]

32. **object (NSG)**
Representation of physical objects and the activities of physical objects of interest to intelligence consumers, such as equipment, facilities, organizations, activities, events, and issues.
[ABI 2017 Concept of Operations Version 4.0 September 2015]

33. **Object-Based Production**
The act of organizing intelligence around **objects** (32) of interest – not by collection or data type – which allows users to aggregate intelligence information from across multiple intelligence domains and associate it with particular objects of interest.
[ABI 2017 Concept of Operations Version 4.0 September 2015]

34. **operation**
specification of a transformation or query that an object may be called to execute
[ISO 19119:2005, definition 4.3]

35. **operation (UML)**
a specification of a transformation or query that an object may be called to execute. It has a name and a list of parameters.
[The Unified Modelling Language Reference Manual, 1999]

36. **property**
facet or attribute of an object, referenced by a name
[OGC 09-026r214-103, definition 4.21]

37. **resource**
asset or means that fulfils a requirement
[OGC 09-026r214-103, definition 4.23]
NOTE In this International Standard, the resource is a **feature** (19), or any identifiable component of a feature (e.g. a property of a feature)

38. **remote resource**
resource (37) that is not under direct control of a system
NOTE In this International Standard, the system is a web feature service. The resource is not held in any data store that is directly controlled by that service and thus cannot be directly retrieved by the service.

39. **request**
invocation of an **operation** (34) by a **client** (14)
[ISO 19128:2005, definition 4.10]

40. **relocate**
<reference> update a reference to a resource that has been moved or copied to a new location
EXAMPLE A **server** (46) is generating a **response** (42) to a GetFeature **request** (39), it has to copy a referenced **feature** (19) into the response document and the server has to "relocate" the original link contained in the referencing feature to the copy placed in the response document.

41. **resolve**
retrieval of a referenced resource and its insertion into a server-generated response document
NOTE The insertion may be accomplished by either replacing the reference in-line with a copy of the resource or by relocating the reference to point to a copy of the resource that has been placed in the response document.

42. **response**
result of an **operation** (34) returned from a **server** (46) to a **client** (14)
[ISO 19128:2005, definition 4.11]

43. **response model**
schema (44) defining the properties of each **feature** (19) type that can appear in the **response** (42) to a query **operation** (34)
NOTE This is the schema of feature types that a **client** (14) can obtain using the DescribeFeatureType operation.

44. **schema**
formal description of a model
[ISO 19101:2002, definition 4.25]
NOTE In general, a schema is an abstract representation of an object's characteristics and relations to other objects. An XML schema represents the relationship between the **attributes** (1) and **elements** (18) of an XML object (for example, a document or a portion of a document).

45. **schema <XML Schema>**
collection of **schema** (44) components within the same target **namespace** (30)
[ISO 19136:2007, definition 4.1.54]
EXAMPLE Schema components of W3C XML Schema are types, **elements** (18), **attributes** (6), groups, etc.

46. **server**
particular instance of a **service** (47)
[ISO 19128:2005, definition 4.12]

47. **service**
distinct part of the functionality that is provided by an entity through **interfaces** (22)
[ISO 19119:2005, definition 4.1]

48. **service metadata**

metadata describing the **operations** (34) and geographic information available at a **server** (46)
[ISO 19128:2005, definition 4.14]

49. standards conformance testing (GFMSA)
testing performed to determine the extent to which a system or subsystem adheres to or implements a standard. It involves testing the capabilities of an implementation against both the conformance requirements in the relevant standard(s) and the statement of the implementation's capabilities.

50. System Under Test (SUT) (GFMSA)
the computer hardware, software and communication network required to support an IUT.

51. Structured Observation Management
NGA's approach to capture, store, standardize, and serve content from primary GEOINT sources, decreasing the time required for analysts to search for exploited data and focus on qualitative and quantitative analysis.
[ABI 2017 Concept of Operations Version 4.0 September 2015]

52. Traversal <XML>
using or following an XLink link for any purpose
[W3C XLink:2001]

53. tuple
ordered list of values
[ISO 19136:2007, definition 4.1.63]
NOTE In this International Standard, the order list will generally be a finite sequence for **features** (4.7), each of a specific feature type.

54. Uniform Resource Identifier
unique identifier for a resource, structured in conformance to IETF RFC 2396
[ISO 19136:2007, definition 4.1.65]
NOTE The general syntax is <scheme>::<scheme-specified-part>. The hierarchical syntax with a namespace (4.16) is <scheme>://<authority><path>?<query>

3.2 Abbreviations

ABAC	Attribute Based Access Control
ABI	Activity Based Intelligence
ATS	Abstract Test Suite
CRS	Coordinate Reference System
DCP	Distributed Computing Platform
DGIWG	Defence Geospatial Information Working Group
DoD	Department of Defense
EPSG	European Petroleum Survey Group[1]

[1] The EPSG was absorbed into the International Association of Oil & Gas Producers (IOGP) in 2005. The EPSG Geodetic Parameter Dataset is now maintained by the Geodesy

GFMSA	GEOINT Functional Manager Standards Assessment program
GML	Geography Markup Language
HTTP	Hypertext Transfer Protocol
IC	Intelligence Community
ICS	Implementation Conformance Statement
IUT	Implementation Under Test
GEOINT	Geospatial-Intelligence
GUIDE	Globally Unique Identifiers for Everything (ODNI)
IDAM	Identity and Access Management
IETF	Internet Engineering Task Force
ISM	Information Security Marking (ODNI standard)
ISO	International Organization for Standardization
ISO/DIS	ISO Draft International Standard
ITU	International Telecommunication Union
ITU-T	Telecommunication Standardization Sector of ITU
JSON	JavaScript Object Notation
JSON-LD	JavaScript Object Notation for Linked Data
KVP	Keyword Value Pairs
MIME	Multipurpose Internet Mail Extension
NAS	NSG Application Schema
NGA	National Geospatial-Intelligence Agency
NoSQL	Not Only SQL
NSG	National System of Geospatial-Intelligence
NSS	National Security Systems
ODNI	Office of the Director of National Intelligence
OGC	Open Geospatial Consortium
OWS	OGC Web Service
PDP	Policy Decision Point

Subcommittee of the IOGP Geomatics Committee. The EPSE acronym was retained by industry for use with the geodetic parameters.

RFC	Request for Comments (IETF)
SOAP	Simple Object Access Protocol
SOM	Structured Observation Management
SPARQL	Simple Protocol And RDF Query Language
SQL	Structured Query Language
SRS	Spatial Reference System
SUT	System Under Test
UML	Unified Modelling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
UPS	Universal Polar Stereographic
UTM	Universal Transverse Mercator
UUID	Universally Unique Identifier
WFS	Web Feature Service
WGS	World Geodetic System
WSDL	Web Services Description Language
XML	Extensible Markup Language
XPath	XML Path Language
XQuery	XML Query

Table 1 : Abbreviations

4 Conformance

4.1 Profiles and Conformance

ISO 19106:2004 *Geographic information - Profiles* defines two classes of conformance, which may be generally thought of as profile types. Conformant Class 1 profiles are a pure subset of the ISO geographic information standards. Conformant Class 2 profiles have the same basis as Class 1 but include extensions within the contexts permitted by the base standard. Additionally, a Class 2 profile permits the profiling of non-ISO geographic information standards as part of the profile.

The same approach is applied to the creation of NSG standards. NSG-specific requirements are identified, then the base standard is restricted (subset) or extended to address those requirements. This approach minimizes incompatibilities between profiles of the base standard while allowing the profiles to accommodate community-specific requirements.

This NSG Profile of the WFS is a Class 2 profile (defines a set of extensions and restrictions) of the DGIWG Web Feature Service 2.0 Profile. The DGIWG Profile, in turn, is a Class 2

profile of version 2.0.2 of the OGC Web Feature Service standard [OGC 09-025r2]. The Web Feature Service incorporates version 2.0.2 of the OGC Filter Encoding standard [OGC 09-026r2].

4.2 Conformance Classes

This NSG Profile is organized into Conformance Classes. Each Conformance Class is specified though a set of requirements. Conformance to a Conformance Class is achieved when all of the requirements for that conformance class are met.

A mapping of Conformance Classes to top-level requirements is provided in Table 2.

Requirements are stated once within the body of this specification. A requirement is then referenced using its' requirement number (ex. R22 = requirement 22). Each requirement statement is followed by accompanying explanatory text. Paragraphs which are explicitly referenced by a requirement shall be considered a part of that requirement. A mapping of the requirements to their defining paragraphs is provided in Table 4.

Many of the requirements in this Profile reference other requirements. Satisfaction of a requirement is only achieved if all dependent requirements are satisfied. Annex A provides a mapping between each requirement and the tests that must be passed to satisfy that requirement. The tests form a tree structure. A "root" test may identify "child" tests. Child tests may identify their own child tests. A root test is not passed until all of its child tests have been passed. Conformance testing should begin with the Conformance Class requirements from Table 2 and work down the dependency tree from there.

Conformance Class	Description	Requirements
NSG Basic WFS	The service shall comply with the DGIWG Basic WFS conformance class as extended and restricted by this specification.	NSG R1
NSG Locking WFS	The service shall comply with the DGIWG Locking (Transactional) WFS conformance class as extended and restricted by this specification.	NSG R1, R2
Inheritance	The service shall implement the schema-element() function for XPath expressions.	NSG R1, R4
Remote Resolve	The service shall implement the ability to resolve remote resource references.	NSG R1, R5
NSG Manage Stored Queries	The service shall implement the CreateStoredQuery and the DropStoredQuery operations.	NSG R1, R3
SOAP	The service shall implement XML encoded requests and results within SOAP Envelopes	NSG R1, R6
Enhanced Paging	The service shall implement the ability to return the page specified by the user from a set of response features or values.	NSG R33, R34

Table 2: NSG Profile Conformance Class Requirements

The NSG Conformance Classes are derived from the Conformance Classes defined in the OGC standards and the DGIWG WFS Profile. The relationships between these classes are illustrated in Table 3.

NSG Conformance Class	DGIWG Conformance Class	OGC Conformance Class
NSG Basic WFS	DGIWG Basic WFS Requirement #1 Requirement #12 Requirement #13	Simple WFS
		Basic WFS
		HTTP GET
		HTTP POST

	DGIWG Recommendation #3	Standard Joins
	DGIWG Recommendation #4	Spatial Joins
	DGIWG Recommendation #5	Temporal Joins
	DGIWG Recommendation #6	Response Paging
	Not required (Section 6.4)	Feature Versions
NSG Locking WFS	DGIWG Locking (Transactional) WFS	Transactional WFS
		Locking WFS
Inheritance	Not required (Section 9.5)	Inheritance
Remote Resolve	Not recommended (Section 9.4)	Remote Resolve
NSG Manage Stored Queries	Not required (Section 7.2.5)	Manage Stored Queries
SOAP	Not required (Sections 7.5 and 8.4)	SOAP
Enhanced Paging	NA	NA

Table 3 : Conformance Class Relationships

Many of the OGC Conformance Classes were made mandatory in either the DGIWG or NSG Profiles. Related mandatory classes are aggregated together into a single Conformance Class. The seven NSG Conformance Classes incorporate all fifteen of the OGC Conformance Classes.

4.3 Conformance Validation

4.3.1 NSG Directive 3201

NSG Directive 3201 establishes the National System for Geospatial Intelligence (NSG) GEOINT Functional Manager Standards Assessment (GFMSA) program. The GFMSA defines and implements NSG methods and processes to assess and assert Information Technology (IT) and National Security Systems (NSS), referred to as IT, conformance to GEOINT data and service standards within the Department of Defense (DoD) and/or the Intelligence Community (IC). Standards conformance is a critical contributing factor in attaining interoperability based on achievement of validated capability requirements.

4.3.2 Implementation Under Test (IUT)

Those planning to test conformance should identify the nature of the IUT and the functional and/or operational risk of undetected conformance defects when selecting a conformance test approach. This includes identifying appropriate test points for data produced or consumed, or for web service and client interactions. In some cases, the IUT could include profiles, implementation specifications, and/or products (format, content, compression and/or metadata) in addition to hardware and/or software implementations.

Various conformance test approaches may be suitable for the circumstances of a specific IUT. Test approaches may include one or more of the following: documentation review, incorporation of branded or conformance registered products, demonstration, evaluation, surveys, questionnaires, manual inspection of results, comparison to sample data, direct testing (use of automated test engines, test harnesses, test scripts, and other automated or semi-automated tools).

4.3.3 Reference Implementation

A Reference Implementation (RI) is a conformant, trusted or well-known exemplar implementation of one or more standards used to support standards conformance and interoperability testing. It also supports standards conformant development by providing example source code for concepts described in the standards. In some instances, the RI

may be suitable for reuse by developers in their own development of the standardized function or service.

Reference implementations of OGC standards are available from the OGC Reference Implementation page at https://cite.opengeospatial.org/reference and from the OGC GitHub at https://github.com/opengeospatial/cite/wiki/Reference-Implementations. Currently there are three reference implementations of WFS 2.0.

A reference implementation for this Profile was developed in OGC Testbed 12. However, that RI was based on a draft of the NSG Profile so it is not suitable for public use. Efforts to update and validate the existing RI and to promote development of additional RIs will commence once Version 1.0.0 of this Profile is finalized.

4.3.4 Conformance Test Resources

XML Schema

OGC standards are designed to be somewhat agnostic to the implementing technology. This allows incorporation of new technologies as they become available. The dominant implementation technologies today are Web Services and XML. Therefore, the OGC supports their standards by providing XML schema as a resource for implementers. OGC XML schemas are available from http://schemas.opengis.net/.

XML schema for the DGIWG profile of WFS 2.0 have been developed but are not yet publically available.

The XML schema for this Profile will be an extension of the OGC and DGIWG schema. Efforts to develop these schema and make them publicly available will commence once Version 1.0.0 of this Profile is finalized.

Conformance Tests

Conformance testing for this Profile will be supported by the OGC Team Engine (http://cite.opengeospatial.org/teamengine/). Team Engine currently supports a compliance test for WFS 2.0. This test will be extended to support the additional requirements introduced by the NSG Profile. Work on these extensions will commence once Version 1.0.0 of this Profile is finalized.

5 Requirements

The purpose of a Profile is to extend and restrict a standard to address community-specific requirements. The NSG requirements addressed in this Profile were identified through interaction with Feature content stakeholders including:

- NGA Consolidated Libraries (NCL) Program Management Office (PMO) and developers.
- ODNI Content Discovery and Retrieval Standards
- ES&IS Program initiatives – Cloud migration including information assurance
- NGA Content Portfolio – NGA Content strategy
- NGA GLOBE API development effort – User of NCL interfaces
- Army Geospatial Center – DoD tactical requirements

The requirements (better described as needs) are as follows:

Profile the DGIWG Profile

"The WFS 2.0 International Standard specifies the behavior of a service that provides transactions on and access to geographic features. It specifies discovery operations, query operations, locking operations, transaction operations and operations to manage stored parameterized query expressions. The International Standard states that a service supporting a specific operation shall be capable of handling any well-formed request."[2]

The DGIWG WFS Profile requires conformance to OGC 09-025r2 and mandates specific operations to be supported by the service. The Profile also describes how the service shall respond. In other words it identifies the set of mandatory elements a service shall provide.

The NSG WFS Profile extends and restricts the DGIWG WFS Profile as required to address NSG specific requirements.

CDR Compatibility

NSG services reside in an environment which is not exclusively GEOINT. Many non-GEOINT users will need to access WFS content using other discovery and access standards. This Profile extends and restricts the DGIWG Profile to facilitate integration with those standards. In particular, the ODNI mandated Content Discovery and Retrieval (CDR) standards.

Information Assurance

NSG services operate in an environment with strict information assurance requirements. These requirements exceed those encountered by commercial systems. This Profile provides some of the additional features needed to meet those requirements.

Time-Band Versioning

Deployed DoD units require a large amount of geospatial data and have limited bandwidth with which to access it. Therefore, their Standard Operating Procedure (SOP) is to only download data which is more recent than what they currently have. OGC and ISO standards provide support for this capability. But that support is optional and somewhat ambiguous. The NSG Profile of the WFS addresses this shortfall.

Requirements	Paragraph
Requirement 1: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall demonstrate conformance to the DGIWG Basic WFS Conformance Class.	***6.2***
Purpose: DGIWG Profile Compliance	
Requirement 2: A WFS Service that claims to be conformant to the NSG Locking WFS Conformance Class shall demonstrate conformance to the NSG Basic WFS and DGIWG Locking WFS Conformance Classes.	***6.2***
Purpose: DGIWG Profile Compliance	
Requirement 3: A WFS Service that claims to be conformant to the NSG Manage Stored* Queries *Conformance Class shall demonstrate conformance to the NSG Basic	***6.2***

[2] DGIWG 122, DGIWG – Web Feature Service 2.0 Profile, (February 2015)

Requirements	Paragraph
WFS Conformance Class and with the Manage Stored Queries Conformance Class from the OGC WFS standard. **Purpose: DGIWG Profile Compliance**	
Requirement 4: A WFS Service that claims to be conformant to the NSG Inheritance WFS Conformance Class shall demonstrate conformance to the NSG Basic WFS Conformance Class and with the Inheritance Conformance Class from the OGC WFS standard. **Purpose: DGIWG Profile Compliance**	***6.2***
Requirement 5: A WFS Service that claims to be conformant to the NSG Remote Resolve WFS Conformance Class shall demonstrate conformance to the NSG Basic WFS Conformance Class and with the Remote Resolve Conformance Class from the OGC WFS standard. **Purpose: DGIWG Profile Compliance**	***6.2***
Requirement 6: A WFS Service that claims to be conformant to the NSG SOAP WFS Conformance Class shall demonstrate conformance to the NSG Basic WFS Conformance Class and with the SOAP Conformance Class from the OGC WFS standard. **Purpose: DGIWG Profile Compliance**	***6.2***
Requirement 7: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall implement service bindings as described in section 6.5. ***(expands on WFS 6.4, DGIWG 7.5, and DGIWG 8.4)*** **Purpose: Consolidates multiple requirements in one place**	***6.5***
Requirement 8: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall encode Features as described in section 7.1. ***(expands on WFS 7.1 and DGIWG 6.5)*** **Purpose: Support for recognized alternate encodings**	***7.1***
Requirement 9: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall implement resource identifiers as described in section 9.1 ***(expands on WFS 7.2.1 and 7.2.2)*** **Purpose: Apply ODNI mandated identifiers**	***7.2***
Requirement 10: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall support Resource Versioning as described in section 9.2. This requirement is a restriction on Resource Versioning as described in OGC WFS 2.0 and OGC Feature Encoding 2.0. ***(replaces WFS 7.2.3 and restricts DGIWG 6.4)*** **Purpose: Supports time-band versioning**	***7.3***
Requirement 11: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall implement Predicate Languages as described in section 7.4. ***(expands on WFS 7.4, clarifies DGIWG 7.2.5.1)*** **Purpose: Elaborates on DGIWG Query language support**	***7.4***
Requirement 12: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall implement Exception Reporting as described in section 7.5.	***7.5***

Requirements	Paragraph
(expands on WFS 7.5)	
Purpose: Captures exception codes added or changed through other requirements	
Requirement 13: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall support the "count" Standard Presentation Parameter.	***7.6.1***
(modifies WFS 7.6.3)	
Purpose: CDR compatibility	
Requirement 14: The default value of the "count" Standard Presentation Parameter shall be 10.	***7.6.1***
(modifies WFS 7.6.3)	
Purpose: CDR compatibility	
Requirement 15: A WFS Service that claims to be conformant to the NSG Basic WFS Profile may implement the "timeout" extension to the Standard Presentation Parameters. If implemented, this parameter shall comply with the following requirements:	***7.6.2***
- The timeout parameter shall specify the maximum clock time the WFS should spend processing the request.	
- The units of the timeout parameter shall be seconds.	
- Upon expiration of the specified timeout period, the WFS shall terminate the request.	
- Upon expiration of the specified timeout period, the WFS shall return an OperationProcessingTimeout error to the client.	
(extends WFS 7.6.3)	
Purpose: CDR compatibility	
Requirement 16: A WFS Service that claims to be conformant to the NSG Basic WFS Profile shall support the mandatory formats and parameter values as described in Table 7 for the outputFormat parameter of the StandardPresentationParameters. This support shall be in addition to the formats requirement in WFS 7.6.3.7	***7.6.3***
(modifies WFS 7.6.3)	
Purpose: Support for recognized alternate encodings	
Requirement 17: A WFS Service that claims to be conformant to the NSG Basic WFS Profile may support an additional ResultType value of "index". If supported, the index result type shall be implemented as described in Section 7.6.4	***7.6.4***
(modifies WFS 7.6.3.1)	
Purpose: CDR compatibility	
Requirement 18: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall implement the paging operations described in WFS Section 7.7.4.4.	***7.7***
(restricts WFS 2 and DGIWG Recommendation 6)	
Purpose: CDR compatibility	

Requirements	Paragraph
Requirement 19: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class and supports the "index" result type, shall include the resultSetID element in StandardResponseParameters. ***(extends WFS 7.7)*** **Purpose: CDR compatibility**	***7.7***
Requirement 20: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall demonstrate conformance to the requirements in Section 6 in their implementation of the GetCapabilities operation. ***(modifies WFS 8 and DGIWG 7.2.1)*** **Purpose: Allocate common requirements to the GetCapabilities operation**	***8.1***
Requirement 21: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall clearly identify the profile used by the WFS service by including at least the following statement: ***<Abstract>This service implements the NSG BASIC WFS profile of WFS 2.0</Abstract>.*** ***(modifies DGIWG Requirement #2)*** **Purpose: Advertise the standard and version implemented by the service**	***8.1.3.1.1***
Requirement 22: A WFS Service that claims to be conformant to the NSG Locking WFS Conformance Class shall clearly identify the profile used by the WFS service by including at least the following statement: ***<Abstract>This service implements the NSG LOCKING WFS profile of WFS 2.0</Abstract>.*** ***(modifies DGIWG Requirement #16)*** **Purpose: Advertise the standard and version implemented by the service**	***8.1.3.1.1***
Requirement 23: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall provide one of the following profile designations: ***- <Profile>http://www.nga.mil/service/wfs/2.0/profile/basic</Profile>*** ***- <Profile>urn:nga:service:wfs:2.0:profile:basic</Profile>*** ***(modifies DGIWG Recommendation #2)*** **Purpose: Advertise the standard and version implemented by the service**	***8.1.3.1.6***
Requirement 24: A WFS Service that claims to be conformant to the NSG Locking WFS Conformance Class shall provide one of the following profile designations: ***-*** ***<Profile>http://www.dgiwg.org/service/wfs/2.0/profile/locking</Profile>*** ***- <Profile>urn:dgiwg:service:wfs:2.0:profile:locking</Profile>*** ***(modifies DGIWG Recommendation #7)*** **Purpose: Advertise the standard and version implemented by the service**	***8.1.3.1.6***
Requirement 25: A WFS Service that claims to be conformant to the NSG Basic WFS Profile shall identify the highest classification level of the content accessible through the WFS service by populating the <ows:AccessConstraints>.	***8.1.3.1.8***

Requirements	Paragraph
- The classification level shall be encoded using the most recent version of the IC Information Security Marking standard (IC.ISM) ***- The classification level encoded in the IC.ISM markings shall be sufficiently restricted to dominate any content which may reside on the service.*** ***(restricts DGIWG Requirement #4)*** **Purpose: Information Assurance**	
Requirement 26: A WFS Service that claims to be conformant to the NSG WFS Profile shall populate the Service Constraints of the OperationsMetadata document as described in Section 8.1.3.3.1. ***(modifies WFS 8.3.5.3)*** **Purpose: Reflect changes due to other requirements**	***8.1.3.3.1***
Requirement 27: A WFS Service that claims to be conformant to the NSG WFS Profile shall populate the Parameters element of the OperationsMetadata document as described in Section 8.1.3.3.2. ***(modifies WFS 8.3.5.3)*** **Purpose: Reflect changes due to other requirements**	***8.1.3.3.2***
Requirement 28: A WFS Service that claims to be conformant to the NSG WFS Profile shall populate the Operation Constraints of the Operation Metadata document as described in Section 8.1.3.3.4. ***(modifies WFS 8.3.5.3)*** **Purpose: Reflect changes due to other requirements**	***8.1.3.3.4***
Requirement 29: A WFS Service that claims to be conformant to the NSG WFS Profile shall populate the Parameters element of the Operation element as described in Section 8.1.3.3.5. ***(modifies WFS 8.3.5.3)*** **Purpose: Reflect changes due to other requirements**	***8.1.3.3.5***
Requirement 30: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall demonstrate conformance to the requirements in Sections 7.1, 7.2, and 7.5 in their implementation of the DescribeFeatureType operation. ***(modifies WFS 10 and DGIWG 7.2.4)*** **Purpose: Allocate common requirements to the DescribeFeatureType operation**	***8.2***
Requirement 31: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall demonstrate conformance to the requirements in Sections 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7, 7.8.1, and 7.8.2 in their implementation of the GetPropertyValue operation. ***(modifies WFS 10 and DGIWG 7.2.4)*** **Purpose: Allocate common requirements to the GetPropertyValue operation**	***8.3***
Requirement 32: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall demonstrate conformance to the requirements in Sections 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7, 7.8.1, and 7.8.2 in their implementation of the GetFeature operation.	***8.4***

Requirements	Paragraph
(modifies WFS 11 and DGIWG 7.2.3)	
Purpose: Allocate common requirements to the GetFeature operation	
Requirement 33: A WFS Service that claims to be conformant to the NSG Enhanced Paging Conformance Class shall implement the PageResults operation as described in Section 8.5.	***8.5***
(extends DGIWG WFS Profile)	
Purpose: CDR compatibility	
Requirement 34: A WFS Service that claims to be conformant to the NSG Enhanced Paging Conformance Class shall demonstrate conformance to the requirements in Sections 7.1, 7.2, 7.3, 7.5, 7.6, and 7.7 in their implementation of the PageResults operation.	***8.5***
(extends DGIWG WFS Profile)	
Purpose: Allocate common requirements to the PageResults operation	
Requirement 35: A WFS Service that claims to be conformant to the NSG Locking WFS Conformance Class shall demonstrate conformance to the requirements in Sections 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7, 7.8.1, and 7.8.2 in their implementation of the LockFeature operation.	***8.6***
(modifies WFS 12 and DGIWG 8.2.3)	
Purpose: Allocate common requirements to the LockFeature operation	
Requirement 36: A WFS Service that claims to be conformant to the NSG Locking WFS Conformance Class shall demonstrate conformance to the requirements in Sections 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7, 7.8.1, and 7.8.2 in their implementation of the GetFeatureWithLock operation.	***8.7***
(modifies WFS 13 and DGIWG 8.2.3)	
Purpose: Allocate common requirements to the GetFeatureWithLock operation	
Requirement 37: A WFS Service that claims to be conformant to the NSG Locking WFS Conformance Class shall demonstrate conformance to the requirements in Sections 7.1, 7.2, 7.3, 7.4, 7.5, 7.8.1, and 7.8.2 in their implementation of the Transaction operation.	***8.8***
(modifies WFS 5 and DGIWG 8.2.2)	
Purpose: Allocate common requirements to the Transaction operation	
Requirement 38: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall demonstrate conformance to the requirements in Sections 7.2 and 7.5 in their implementation of the ListStoredQueries operation.	***8.9***
(modifies WFS 14.3 and DGIWG 7.2.5)	
Purpose: Allocate common requirements to the ListStoredQueries operation	
Requirement 39: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall demonstrate conformance to the requirements in Sections 7.2, 7.3, 7.4, 7.5, 7.8.1, and 7.8.2 in their implementation of the DescribeStoredQueries operation.	***8.10***
(modifies WFS 14.4 and DGIWG 7.2.5)	
Purpose: Allocate common requirements to the DescribeStoredQueries operation	

Requirements	Paragraph
Requirement 40: A WFS Service that claims to be conformant to the NSG Manage Stored Queries Conformance Class shall demonstrate conformance to the requirements in Sections 7.2 and 7.5 in their implementation of the DropStoredQuery operation. ***(modifies WFS 14.6)*** **Purpose: Allocate common requirements to the DropStoredQueries operation**	***8.11***
Requirement 41: A WFS Service that claims to be conformant to the NSG Manage Stored Queries Conformance Class shall demonstrate conformance to the requirements in Sections 7.1, 7.2, 7.3, 7.4, 7.5, 7.8.1, and 7.8.2 in their implementation of the CreateStoredQuery operation. ***(modifies WFS 14.5)*** **Purpose: Allocate common requirements to the CreateStoredQueries operation**	***8.12***

Table 4: Requirements Trace Matrix

6 Basic Service Elements

6.1 Introduction

This clause describes aspects of a Web Feature Service's behavior that are applicable to the service as a whole.

6.2 Pre-Requisites

The NSG Profile of the WFS is derived from both the OGC WFS standard and the DGIWG profile of that standard. Therefore, at a minimum this Profile must be traceable back to the standards which are profiled.

Requirement 1: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall demonstrate conformance to the DGIWG Basic WFS Conformance Class.

Rationale: The Defence Geospatial Information Working Group (DGIWG) is the multi-national body responsible for geospatial standardization for the defense organizations of member nations. As a member nation, the United States complies with the standards developed by this body. In February 2015 the DGIWG produced their profile of the OGC Web Feature Service 2.0. Therefore, that document serves as the baseline for the NSG Profile of the OGC standard.

Note that the DGIWG Profile prohibits implementation of just the Simple WFS conformance class. The Basic WFS conformance class is the minimal acceptable implementation.

Requirement 2: A WFS Service that claims to be conformant to the NSG Locking WFS Conformance Class shall demonstrate conformance to the NSG Basic WFS and DGIWG Locking WFS Conformance Classes.

Rationale: The DGIWG Profile of the WFS requires that a service which implements the DBIWG Locking Conformance Class also comply with the DGIWG Basic Conformance Class. This requirement carries the DGIWG requirement over into the NSG Profile.

Requirement 3: A WFS Service that claims to be conformant to the NSG Manage Stored Queries Conformance Class shall demonstrate conformance to the NSG Basic WFS Conformance Class and with the Manage Stored Queries Conformance Class from the OGC WFS standard.

Rationale: The DGIWG Profile of the WFS does not include a conformance class for manage stored queries. Therefore, the NSG Profile must be based on the OGC conformance class. That class mandates that at least the Basic conformance class is implemented.

Requirement 4: A WFS Service that claims to be conformant to the NSG Inheritance WFS Conformance Class shall demonstrate conformance to the NSG Basic WFS Conformance Class and with the Inheritance Conformance Class from the OGC WFS standard.

Rationale: The DGIWG Profile of the WFS does not include a conformance class for Inheritance. Therefore, the NSG Profile must be based on the OGC conformance class. Conformance to the NSG Basic class is required for all derivative conformance classes.

Requirement 5: A WFS Service that claims to be conformant to the NSG Remote Resolve WFS Conformance Class shall demonstrate conformance to the NSG Basic WFS Conformance Class and with the Remote Resolve Conformance Class from the OGC WFS standard.

Rationale: The DGIWG Profile of the WFS does not include a conformance class for Remote Resolve. Therefore, the NSFG Profile must be based on the OGC conformance class. Conformance to the NSG Basic class is required for all derivative conformance classes.

Requirement 6: A WFS Service that claims to be conformant to the NSG SOAP WFS Conformance Class shall demonstrate conformance to the NSG Basic WFS Conformance Class and with the SOAP Conformance Class from the OGC WFS standard.

Rationale: The DGIWG Profile of the WFS mandates implementation of the HTTP GET and POST Conformance Classes (Requirements 12 and 13) but does not discuss a SOAP conformance class. Therefore, the NSG Profile must be based on the SOAP conformance class from the OGC WFS standard. Conformance to the NSG Basic class is required for all derivative conformance classes.

6.3 Version Numbering and Negotiation

This Profile makes no changes to version numbering and negotiation as described in the DGIWG and OGC standards.

6.4 Namespaces

A common understanding of the terms used to populate request and response parameters is essential for the effective exchange of information. Universal agreement on these terms is a near-impossible task. Namespaces divide the universe of discourse into domains. Terms are defined within the scope of a Namespace. A Namespace-term pair identifies the specific definition of a term within the scope of that Namespace. The collection of Namespaces is itself a collection of terms. A common understanding of the terms which identify

Namespaces is essential to a common understanding of the terms the Namespaces describe. Table 5 describes the Namespace identifiers used in this specification.

Namespace	Description
http://www.opengis.net/wfs/2.0	WFS interface vocabulary
http://www.opengis.net/gml/3.2	GML vocabulary (see ISO 19136:2007)
http://www.opengis.net/fes/2.0	OGC Filter vocabulary (see ISO 19143, 5.4)
http://www.opengis.net/ows/1.1	OWS Common vocabulary (see OGC- 06121r3)
http://metadata.ces.mil/dse/ns/GSIP/schema/nas	NSG Extensions vocabulary
urn:us:gov:ic:ism	DoD/IC Security Markings
urn:us:gov:ic:ntk	DoD/IC Need To Know Markings
http://www.w3.org/2001/XMLSchema	XML Schema
http://example.org/vocab	Use for examples, not for operational use

Table 5 : NSG WFS Namespaces

6.5 Service Bindings

Requirement 7: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall implement service bindings as described in section 6.5.

(expands on WFS 6.4, DGIWG 7.5, and DGIWG 8.4)

Rationale: The main body of this Profile defines the encoding of WFS request and response messages independent of any particular communication protocol. However, agreement on a communications protocol, or binding, is a pre-condition to interoperability. Table 6 identifies the bindings (DCP) which are mandatory and optional for each WFS operation. The table also specifies the request and response encodings.

Operation	DCP	Mandatory	Request Encoding	Response Encoding
GetCapabilities	HTTP GET	M	KVP	XML
	HTTP POST	M	XML	XML
	SOAP	O	SOAP	XML
DescribeFeatureType	HTTP GET	M	KVP	XML
	HTTP POST	M	XML	XML
	SOAP	O	SOAP	XML
GetPropertyValue	HTTP GET	M	KVP	XML
	HTTP POST	M	XML	XML
	SOAP	O	SOAP	XML
GetFeature	HTTP GET	M	KVP	XML
	HTTP POST	M	XML	XML
	SOAP	O	SOAP	XML
PageResults	HTTP GET	M	KVP	XML
	HTTP POST	M	XML	XML
	SOAP	O	SOAP	XML
GetFeatureWithLock	HTTP GET	M	KVP	XML
	HTTP POST	M	XML	XML
	SOAP	O	SOAP	XML
LockFeature	HTTP GET	M	KVP	NA
	HTTP POST	M	XML	NA
	SOAP	O	SOAP	NA
Transaction	HTTP GET	O	KVP	XML
	HTTP POST	M	XML	XML
	SOAP	O	SOAP	XML
CreateStoredQuery	HTTP GET	O	KVP	NA
	HTTP POST	M	XML	NA
	SOAP	O	SOAP	NA
DropStoredQuery	HTTP GET	M	KVP	NA

	HTTP POST	O	XML	NA
	SOAP	O	SOAP	NA
ListStoredQuery	HTTP GET	M	KVP	XML
	HTTP POST	M	XML	XML
	SOAP	O	SOAP	XML
DescribeStoredQuery	HTTP GET	M	KVP	XML
	HTTP POST	M	XML	XML
	SOAP	O	SOAP	XML

Table 6 : Service Bindings

7 Common Elements

This clause describes aspects of a Web Feature Service's behavior that are applicable to more than one operation or interface. The requirements identified in this section may appear as dependencies to requirements which are specific to an operation or interface.

7.1 Encoding of Features

Requirement 8: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall encode Features as described in section 7.1.

(expands on WFS 7.1 and DGIWG 6.5)

Rationale: The DGIWG Profile identifies eight (8) feature encodings in addition to those described in the OGC WFS standard. This NSG Profile adds a URN value which can be used to indicate a JSON encoding. The full set of feature encodings supported by this NSG Profile are describes in Table 7. Encodings marked as mandatory shall be supported by all implementations. If an encoding is supported, then that encoding shall be designated using the specified value.

Name	Description	Value	Mandatory or Optional
GML 3.2	Minimum requirement of the OGC WFS standard	"application/gml+xml; version=3.2"	M
GML 2	Previous version of GML (DGIWG)	“GML2”	O
GML 2.1.2	Previous version of GML (DGIWG)	"text/xml; subtype=gml/2.1.2"	O
GML 2 zipped	Previous version of GML with zip compression (DGIWG)	“GML2-GZIP”	O
SHAPE zipped	SHAPE file with zip compression (DGIWG)	“SHAPE-ZIP”	O
CSV	Comma-Separated Values (DGIWG)	“csv”	O
GML 3	GML Version 3 (DGIWG)	“gml3”	O
GML 3.2	GML Version 3.2 (DGIWG)	“gml32”	O
JSON	Java Script Object Notation (DGIWG)	“json”	O
JSON	Java Script Object Notation (OGC WFS 2.5)	“application/json”	O

Table 7 : Feature Encodings

7.2 Resource Identifiers

Requirement 9: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall implement resource identifiers as described in section 9.1

(expands on WFS 7.2.1 and 7.2.2)

Rationale: NSG requirements described in this specification, as well as NSG strategies for content management, impose requirements on resource identifiers and their behavior. Section 9.1 addresses how those requirements will be addressed when a Resource Identifier is used with an NSG WFS.

7.3 Resource Versioning

The OGC WFS and Filter Encoding standards support client selection of the version or versions of a resource to be accessed. The OGC standards provide a number of different versioning techniques, none of which are mandatory. This NSG Profile restricts this capability to one versioning technique and makes support of that technique mandatory.

Requirement 10: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall support Resource Versioning as described in section 9.2. This requirement is a restriction on Resource Versioning as described in OGC WFS 2.0 and OGC Feature Encoding 2.0.

(replaces WFS 7.2.3 and restricts DGIWG 6.4)

Rationale: Deployed DoD units require a large amount of geospatial data and have limited bandwidth with which to access it. Therefore, their Standard Operating Procedure (SOP) is to only download data which is more recent than what they currently have. In NSG Need #2587 the U.S. Army imposed a requirement on NGA to establish a method to specify and select on the version of vector, raster file, and raster tile data.

It is desirable that there be one versioning construct for all of the NSG. SOM, ABI, and Big Data Analytics yield a tremendous increase in the volume and velocity of NSG content. A time-stamp based approach to versioning would address these needs as well of those of the DoD. This approach (described in Section 9.2) has the following properties:

- Feature versioning support is mandated
- Feature versioning shall be based on date and time
- Feature versioning shall be compatible with the Filter Encoding standard

7.4 Predicate Expression Languages

Requirement 11: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall implement Predicate Languages as described in section 7.4.

(expands on WFS 7.4, clarifies DGIWG 7.2.5.1)

Rationale: The DGIWG Profile allows five (5) predicate expression languages in addition to those described in the OGC WFS standard. The full set of predicate expression languages supported by this NSG Profile are describes in Table 7. Languages marked as mandatory shall be supported by all implementations. If a language is supported, then that language

shall be designated using the specified MIME type. Details on how these languages should be implemented are provided in Section 7.8.1.

Name	Description	Identifier[1]	Mandatory or Optional
OGC Filter	OGC Filter Encoding Standard as mandated in the OGC and DGIWG standards	urn:ogc:def:queryLanguage: :OGC-WFS::WFSQueryExpression	M
SQL	Structured Query Language. Commonly used by relational databases. By itself it does not have spatial-temporal capabilities.	urn:ogc:def:queryLanguage: sql	O
SPARQL	A query language used to discover linked data.	application/sparql-query	O
XQuery	A query language for XML document stores	urn:ogc:def:queryLanguage: xquery	O
XPath	A query language for selecting nodes from an XML document.	urn:ogc:def:queryLanguage: xpath	O
Java	Java Persistence Query Language is an SQL-like language for use with a Java Persistence store.	urn:ogc:def:queryLanguage: java	O
[1] NOTE: these identifiers have not been adopted by the OGC at the time of publication			

Table 8 : Predicate Expression Languages

7.5 Exception Reporting

Requirement 12: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall implement Exception Reporting as described in section 7.5.

(expands on WFS 7.5)

Rationale: Provide a single integrated set of exception codes and identify operations where each is applicable.

Exception reporting for WFS services is described in section 7.5 of the OGC WFS Standard. The exception codes defined in the OGC OWS Common standard, OGC WFS standard, and this NSG WFS Profile are provided in Table 9. Implementations of this profile of the WFS shall support the exception codes described in Table 9.

Exception Code Value	Meaning of Code	"locator" value
OWS Common 2.0 Codes		
InvalidParameterValue	Operation request contains an invalid parameter value	Name of parameter with invalid value
InvalidUpdateSequence	Value of (optional) updateSequence parameter, in GetCapabilities operation request, is greater than the current value of the service metadata updateSequence number	None, omit "locator" parameter
MissingParmameterValue	Operation request does not include a parameter value	Name of missing parameter

Exception Code Value	Meaning of Code	“locator” value
NoApplicableCode	No other exceptionCode specified by this service and service applies to this exception	None, omit “locator” parameter
OperationNotSupported	Request is for an operation that is not supported by this service	Name of operation not supported
OptionNotSupported	Request is for an option that is not supported by this service	Identifier of option not supported
VersionNegotiationFailed	List of version in “AcceptVersions” parameter value, in GetCapabilities operation request, did not include any version supported by this service.	None, omit “locator” parameter
WFS 2.0 Codes		
CannotLockAllFeatures	A locking request with a lockAction of ALL failed to lock all the requested features.	If the operation includes the optional "handle" parameter, report its value as the value of the "location" parameter
DuplicateStoredQueryIdValue	The identifier specified for a stored query expression is a duplicate.	The "locator" parameter shall contain the value of the duplicate identifier.
DuplicateStoredQuery-ParameterName	This specified name for a stored query parameter is already being used within the same stored query definition.	The "locator" parameter shall list the name of the duplicate stored query parameter.
FeaturesNotLocked	For services that do not support automatic data locking, this exception indicates that a transaction operation is modifying features that have not previously been locked using a LockFeature or GetFeatureWithLock operation.	If the operation includes the optional "handle" parameter report its value as the value of the "location" parameter.
InvalidLockId	The value of the lockId parameter on a Transaction operation is invalid because it was not generated by the service.	The "locator" parameter shall contain the value of the invalid lockId.
InvalidValue	A Transaction has attempted to insert or change the value of a data component in a way that violates the schema of the feature.	The "locator" parameter shall contain the name of the property being incorrectly modified.
LockHasExpired	The specified lock identifier on a Transaction or LockFeature operation has expired and is no longer valid.	The "locator" parameter shall contain the value of the expired lock identifier.
OperationParsingFailed	The request is badly formed and failed to be parsed by the service.	The "locator" parameter shall contain the value of the "handle" parameter if one is available. Otherwise the "locator" parameter shall contain the name of the badly formed operation.
OperationProcessingFailed	An error was encountered while processing the operation.	The "locator" parameter shall contain the value of the "handle" parameter if one is available. Otherwise the "locator" parameter shall contain the name of the operation that failed.

Exception Code Value	Meaning of Code	“locator” value
ResponseCacheExpired	The response cache used to support paging has expired and the results are no longer available.	If the operation includes the optional "handle" parameter, report its value as the value of the "locator" parameter.
NSG Profile Codes		
OperationProcessingTimeout	Processing time for this request has exceeded the maximum specified in the timeout parameter	The "locator" parameter shall contain the value of the "handle" parameter if one is available. Otherwise the "locator" parameter shall contain the name of the operation that failed.

Table 9 : Exception Codes

Not all exception codes are applicable to all operations. Table 10 provides a mapping of each WFS exception code to the operations which may return that code.

Exception Code Value	Relevant Operations	Conformance Class
OWS Common 2.0 Codes		
InvalidParameterValue	All	NSG Basic WFS
InvalidUpdateSequence	GetCapabilities	NSG Basic WFS
MissingParmameterValue	All	NSG Basic WFS
NoApplicableCode	All	NSG Basic WFS
OperationNotSupported	All	NSG Basic WFS
OptionNotSupported	All	NSG Basic WFS
VersionNegotiationFailed	GetCapabilities	NSG Basic WFS
WFS 2.0 Codes		
CannotLockAllFeatures	LockFeature, GetFeatureWithLock	NSG Locking WFS
DuplicateStoredQueryIdValue	CreateStoredQuery	NSG Manage Stored Queries
DuplicateStoredQueryParameterName	CreateStoredQuery	NSG Manage Stored Queries
FeaturesNotLocked	Transaction	NSG Locking WFS
InvalidLockId	Transaction	NSG Locking WFS
InvalidValue	Transaction	NSG Locking WFS
LockHasExpired	Transaction, LockFeature	NSG Locking WFS
OperationParsingFailed	GetCapabilities, DescribeFeatureType, GetFeature, PageResults, GetPropertyValue, ListStoredQueries, DescribeStoredQueries,	All

Exception Code Value	Relevant Operations	Conformance Class
	Transaction. LockFeature, GetFeatureWithLock, DropStoredQuery, CreateStoredQuery	
OperationProcessingFailed	GetCapabilities, DescribeFeatureType, GetFeature, PageResults GetPropertyValue, ListStoredQueries, DescribeStoredQueries, Transaction. LockFeature, GetFeatureWithLock, DropStoredQuery, CreateStoredQuery	All
ResponseCacheExpired	GetFeature (paging) PageResults	NSG Basic WFS Enhanced Paging
NSG Profile Codes		
OperationProcessingTimeout	GetFeature, PageResults GetPropertyValue, GetFeatureWithLock,	NSG Basic WFS Enhanced Paging NSG Locking WFS

Table 10 : Exception Code Mapping

7.6 Standard Presentation Parameters

Standard presentation parameters are used to control how query results are presented in a Response document. These parameters may appear in the GetPropertyValue, GetFeature, PageResults, and GetFeatureWithLock operations. The NSG Profile modifies and extends the parameters defined in the base standards in order to:

1) To better manage the size of received content (critical for applets and bandwidth limited devices).
2) To facilitate integration of WFS with other Enterprise discovery standards

The NSG Standard Presentation Parameter structure is illustrated in Figure 1.

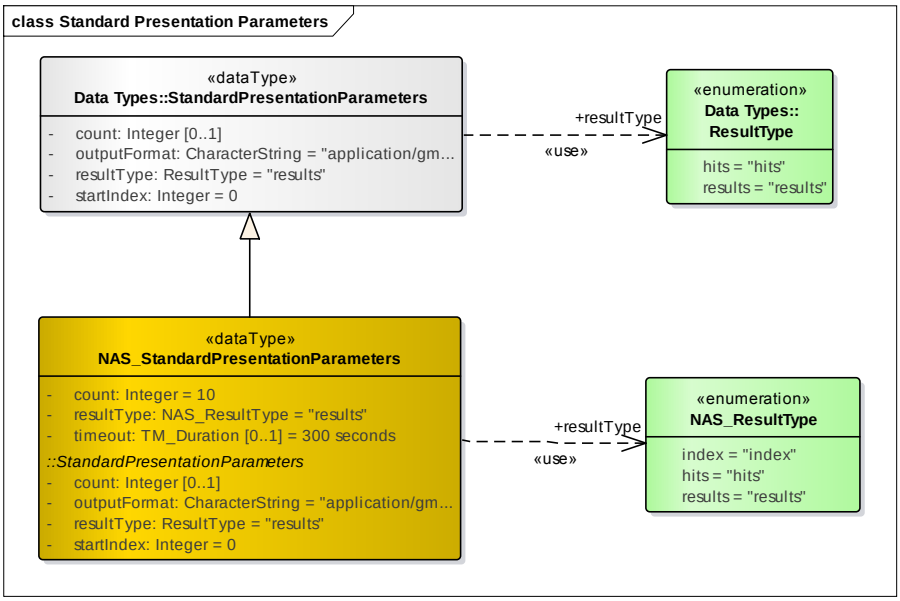

Figure 1 : NSG Standard Presentation Parameters

7.6.1 Count Parameter

Requirement 13: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall support the "count" Standard Presentation Parameter.

(modifies WFS 7.6.3)

Rationale: The "count" parameter imposes a limit on the number of items which can be returned in the result set. This is a critical capability when the client is on a resource-constrained or bandwidth limited platform.

Requirement 14: The default value of the "count" Standard Presentation Parameter shall be 10.

(modifies WFS 7.6.3)

Rationale: This brings the WFS Profile into alignment with the ODNI CDR standards.

7.6.2 Timeout Parameter

Requirement 15: A WFS Service that claims to be conformant to the NSG Basic WFS Profile may implement the "timeout" extension to the Standard Presentation Parameters. If implemented, this parameter shall comply with the following requirements:

- The timeout parameter shall specify the maximum clock time the WFS should spend processing the request.

- The units of the timeout parameter shall be seconds.

- Upon expiration of the specified timeout period, the WFS shall terminate the request.

- Upon expiration of the specified timeout period, the WFS shall return an OperationProcessingTimeout error to the client.

(extends WFS 7.6.3)

Rationale: This is an extension to the WFS standard. WFS provides a timeout for resolving remote references, but not for the request as a whole. This extension allows the client to set a timeout for the entire transaction, a common capability on other discovery standards. In addition it:

a) Protects the WFS from mal-formed queries which could result in a denial of service.

b) Allows the user to specify the query behavior and to take corrective actions when a query gets lost.

7.6.3 OutputFormat Parameter

Requirement 16: A WFS Service that claims to be conformant to the NSG Basic WFS Profile shall support the mandatory formats and parameter values as described in Table 7 for the outputFormat parameter of the StandardPresentationParameters. This support shall be in addition to the formats requirement in WFS 7.6.3.7

(modifies WFS 7.6.3)

Rationale: Non-NSG systems often do not support GML encodings of Feature data. To accommodate these systems, additional encoding formats may be needed. Those additional formats are documented in Table 7.

7.6.4 ResultType Parameter

Requirement 17: A WFS Service that claims to be conformant to the NSG Basic WFS Profile may support an additional ResultType value of "index". If supported, the index result type shall be implemented as described in Section 7.6.4

(modifies WFS 7.6.3.1)

Rationale: The index result type has been added to support incremental refinement of a result set which has been retained by the service. The behavior of an "index" request will be the same as that for a "hits" request except that a result set has been created and can be accessed by the client. The "index" response is an extension of the "hits" response. Those extensions are:

- An identifier for the result set is returned in the resultSetID parameter (see section 7.7).
- A URL to the result set is returned in the "next" response parameter.

The result set is a sequence of Feature Collections, each of which contains "count" Features. The ResultSetID provides a single identifier for the whole result set. Random access to a Feature Collection can be performed using the ResultSetID and PageResults() operation (Section 8.5). The "next" parameter points to the first Feature Collection in the result set. Sequential Navigation through the result set can be performed using the "next" and "previous" parameters as describe in WFS Section 7.7.4.4.

7.7 Standard Response Parameters

Standard response parameters are parameters returned with any whole or partial result set. They provide the client with information they need to understand and access the results. The NSG extension of StandardResponseParameters is illustrated in Figure 2.

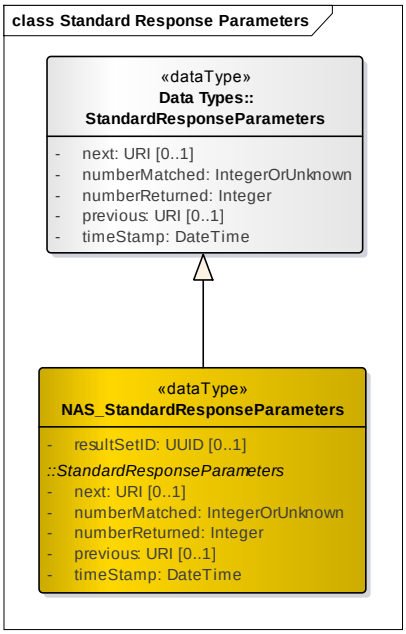

Figure 2 : NAS Standard Response Parameters

> ***Requirement 18: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall implement the paging operations described in WFS Section 7.7.4.4.***
>
> ***(restricts WFS 2 and DGIWG Recommendation 6)***

Rationale: This requirement compliments Requirement R13. The "count" parameter causes the result set to be segmented into pages. Therefore, the response message must support the WFS paging capability to access those pages.

> ***Requirement 19: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class and supports the "index" result type, shall include the resultSetID element in StandardResponseParameters.***
>
> ***(extends WFS 7.7)***

Rationale: This requirement compliments the "index" result type. The resultSetID value can be used to refine the result set through additional operations on that resource. It also supports the PageResults operation by providing a target resource for the PageResults request. Finally, it supports content management by providing an unambiguous global identifier for an NSG resource.

7.8 Query Expressions

Query expressions are governed by three standards, OGC Filter Encoding 2.0.2 (OGC 09-062r2), OGC Web Feature Service 2.0.2 (OGC 09-025r2), and the DGIWG - Web Feature

Service 2.0 Profile (DGIWG 122). This Profile does not introduce any major changes to the query expressions defined by those standards. However, the WFS standards assume that query expressions will be levied against OGC Features. Increasingly this assumption is not valid. This section presents a description of the WFS query expression which is not specific to Feature holdings. Guidance for specific content families (NoSQL, Linked Data, Big Table, etc.) will be provided in a latter version of this Profile.

A query expression defines an action that performs a search over some set of resources and returns a subset of those resources. A query expression can be either a Stored Query or an Ad-Hoc Query. An Ad-Hoc query is a query expression that is fully formed and does not require any prior-knowledge on the part of the queried service. A Stored Query is an Ad-Hoc query that has been stored on the queried service for reuse. All query expressions have an optional Handle property which provides an identifier for that query expression (Figure 3.)

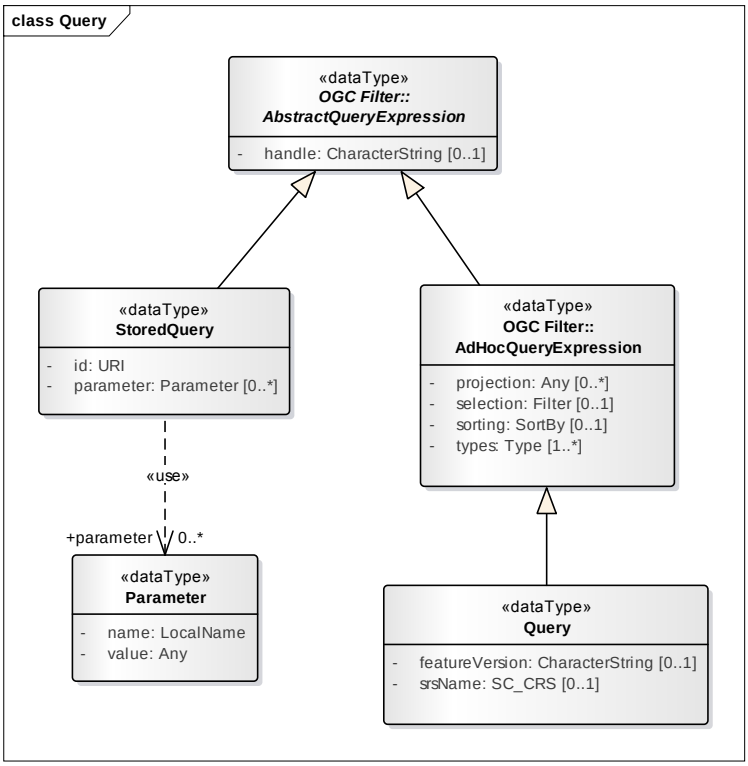

Figure 3 : Query Expressions

7.8.1 Ad-Hoc queries

An ad hoc query expression is a query expression that is fully formed. The queried service is not required to have any prior-knowledge to execute an ad-hoc query. These query expressions have four parts:

1) Types: The names of one or more resource types to query,
2) Projection: Identifies the resource properties to present in the response,
3) Selection: Allows selection of the entities to return based on property values,
4) Sorting: Specifies the order in which the result set is to be presented.

7.8.1.1 Alternate Content Types

While the Web Feature Service was developed to provide OGC Feature data, it is suitable for other content as well. Table 11 provides a translation guide from Feature terminology to analogous concepts from other content models. Implementers of this Profile are free to substitute terms from this table for the OGC Feature terms when appropriate.

> ***Guidance:*** *When specifying requirements from the NSG WFS Profile for a system which serves content which is not OGC Feature data, Program Management Officials may substitute appropriate terms from Table 11 for the corresponding OGC Features terminology.*

Content Model	Namespace	Types	Properties	Returns
OGC Features	Ontology	Feature Type	Feature Properties	Features
Relational Database	NA	Table	Column	Rows
Big Table	NA	Table	Column Family/Column	Rows
Object Database	NA	Class	Attribute	Objects
XML	NA	Document	XML Elements	Document

Table 11 : Query Expression Terminology Mapping

While developed for the Filter Encoding format, a Query Expression can be populated using other query languages as illustrated in Table 12. Some query languages, such as SPARQL, have many additional capabilities. However, they are constrained to the Filter Encoding semantics for this Profile.

Query Language[1]	Term 1	Term 2	Term 3	Term 4
Filter Encoding	Types	Projection	Selection	Sorting
SQL	From	Select	Where	Order By
SPARQL[2]	FROM (DatasetClause)	SELECT (SelectClause)	WHERE (WhereClause)	ORDER (OrderClause)
Java[2]	From_Clause	Select_Clause	Where_Clause	OrderBy_Clause
XQuery	Path to the target XML documents	NA – embedded in the selection	XQuery expression	NA – embedded in the selection
XPath	Path to the target XML documents	NA – embedded in the selection	XPath expression	NA – embedded in the selection

[1]This list of query languages comes from DGIWG Table 6

[2]This mapping is a subset of the full capablities of this query language

Table 12 : Query Language Mapping

The OGC Filter Encoding, SQL, SPARQL, and Java Persistence query languages all follow an SQL encoding style. So they fit neatly into the Types/Projection/Selection/Sorting pattern.

XQuery and Xpath are designed to search XML documents or respositories which can be viewed as XML documents. Since the query expression is constrained by the XML syntax, it cannot be decomposed into the Types/Projection/Selection/Sorting pattern. Therefore, we only have the path to the XML documents to be queried and the query expression to use on the documents.

7.8.1.2 Type Names

These are the names of the resource types to query. At least one is required but there is no restriction on the number of additional resources identified. If you enter more than one type name it implies a join of the results returned from each resource.

A type name can also have an alias. An alias is an alternate name for the resource. For example, to query for NAS airports and roads you can enter:

<wfs:Query typeNames="nas:LandAerodrome nas:Road" aliases="Airport Road">

The ordering of the Type Names is critical. The query expression does not provide an explicit association between Type Names and aliases. The number and order of the aliases must match that of the Type Names.

7.8.1.3 Projection

"Every feature representation generated by a WFS shall include all the mandatory properties for the feature type according to the schema description (see Clause 9), and then may include a selection of the other properties, according to the schema description.

The projection clause enumerates which of the non-mandatory properties of a feature shall be included in the response to a query." (WFS 7.9.2.4.5.1)

In other words, a WFS will only return the properties which are mandatory under the WFS data model. If you want more, you have to ask for them. The Projection clause is how you ask for them.

A property name is the qualified name (see section **Error! Reference source not found.**) or the property. Including a property name informs the service that you wish that values for that property are returned in the result set.

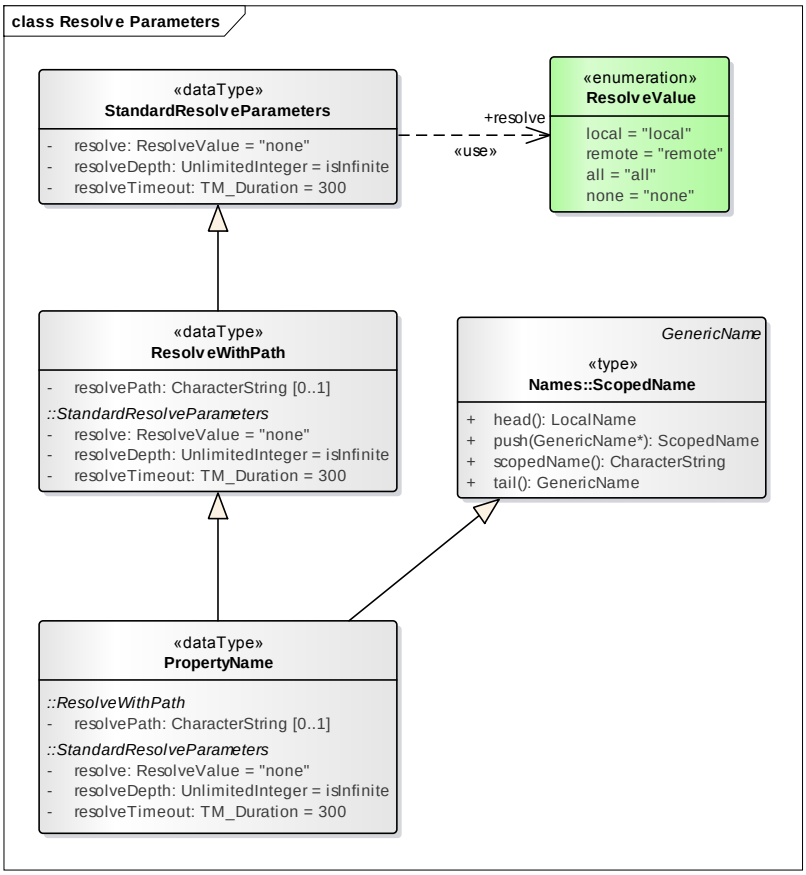

Figure 4: Standard Resolve Parameters

The values of a property can be either data or a reference to where the data may be found. If the property value is a reference, then that reference must be followed to retrieve the actual data.

A reference may also be a sequence of references such as A->B->C->D. Processing a chain of references can be very time and resource intensive. The Standard Resolve Parameters address this by allowing the user to restrict how references are resolved. Those parameters are summarized in Table 13.

Parameter	Effect
Resolve	Indicates how references should be processed: "local" = only follow local references "remote" = only follow remote references "all" = follow any reference "none" = don't follow any references
ResolveDepth	Indicates the maximum number of hops to process. In the case of the A->B->C->D example, a resolveDepth of 2 would cause the service to stop resolving references at C.
ResolveTimeout	Provides a time limit for resolve processing. If the data is not located by the time the timeout expires, then only the initial reference is returned.

Table 13 : Standard Resolve Parameters

In addition to the standard resolve parameters, the WFS standard adds the resolvePath parameter. The normal behavior of the resolve parameter, when its value is not set to "none", is to resolve all resource references to the depth specified by the resolveDepth parameter. If the resolvePath parameter is populated, then resource resolution will only proceed along the specified property path.

7.8.1.4 Selection

A user usually doesn't want to see all of the resources of the selected types. Therefore, they frequently filter the resources based on the values of their properties. The Selection clause is used to identify a subset of resources from a collection of resources whose property values satisfy a set of logically connected predicates. If the property values of a resource satisfy all the predicates, then that resource is included in the result set.

Support for the OGC Filter Encoding language is required but additional languages can be used as illustrated in Table 12.

7.8.1.5 Sort

The sorting clause can be used to assert the order in which resources appear in response to a query expression. This clause is a list of property names with associated indicators that sorting should be ascending or descending. Sorting is done in-order from the first property name to the last.

7.8.1.6 SRS Name

The optional srsName attribute may be used to assert that a specific WFS-supported Coordinate Reference System (CRS) transformation should be applied to the geometries of the features returned in a response document. If no srsName value is supplied, then the feature geometries shall be encoded in the response document using the wfs:DefaultCRS value advertised in the service metadata.

This Profile makes no changes to the SRS Name as defined in the OGC WFS standard. The DGIWG Profile provides one additional requirement:

> *"Requirement 21: A service implementing the DGIWG WFS profiles shall provide their data in:*
> - *CRS:84 WGS84 geographic longitude, then latitude, expressed in decimal degrees*
> - *EPSG:4326 WGS84 geographic latitude, then longitude, expressed in decimal degrees*
> - *Optionally data may be provided in additional CRSs, depending on national requirements."*

and one recommendation:

> *"Recommendation 8: The followings CRSs should be supported as well:*
> - *World Mercator projection, known as EPSG:3395.*
> - *All projections for which validity zone overlaps data published by the service*
>   - *UTM projections over WGS84 (north zones), EPSG:32601 to EPSG:32660*
>   - *UTM projections over WGS84 (south zones), EPSG:32701 to EPSG:32760*
>   - *UPS projection over WGS84 (north zone), EPSG:32661*

- *UPS projection over WGS84 (south zone), EPSG:32761"*

7.8.1.7 Feature Version

The OGC WFS standard includes a FeatureVersion parameter of type CharacterString. It provides no direction on how this parameter should be used.

NSG implementations of the WFS should perform version filtering as described in Section 9.2. This approach does not require use of the FeatureVersion parameter and is in-line with the approach presented in the DGIWG Profile.

This Profile does not prohibit the use of the FeatureVersion parameter, but it is not a recommended practice.

7.8.2 Stored Queries

Good query expressions can be difficult to construct. Stored queries allow a query to be built once then re-used over and over again. A stored query is a subclass of AbstractQueryExpression (Figure 5) and can be used anywhere an ad-hoc query expression is used.

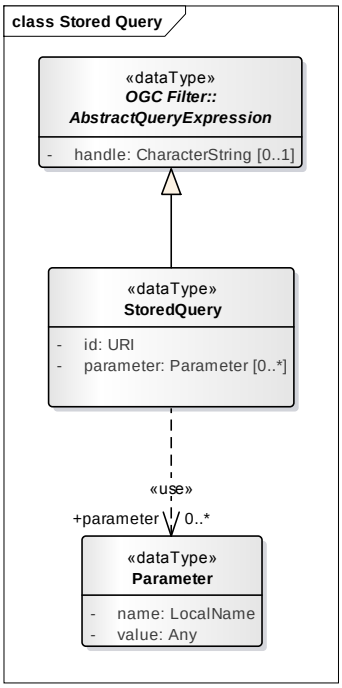

Figure 5 : Stored Query

A stored query consists of the query identifier and an optional set of parameters for that query. The identifier is unique for a single stored query expression. The parameters are provided by the query expression to allow a user to tailor the expression for their needs.

7.8.2.1 Parameters

Stored query expressions are just like ad-hoc query expressions with one difference. A stored query may include the ${<parameter>} token. This token indicates that "${<parameter>}" should be replaced by the value passed in the <parameter> value. For example, if a stored query contained the envelope:

```
<gml:Envelope srsName="urn:fes:def:crs:EPSG::4326">
    <gml:lowerCorner>${lower_corner}</gml:lowerCorner>
    <gml:upperCorner>${upper_corner}</gml:upperCorner>
</gml:Envelope>
```

and the parameters passed were:

lower_corner = “50 40“
upper_corner = “100 60“

Then the query that is executed will contain the envelope:

```
<gml:Envelope srsName="urn:fes:def:crs:EPSG::4326">
    <gml:lowerCorner>50 40</gml:lowerCorner>
    <gml:upperCorner>100 60</gml:upperCorner>
</gml:Envelope>
```

7.8.2.2 Stored Query Descriptions

A stored query expression is difficult to manage, discover, and use by itself. It needs supporting metadata. The metadata which supports a stored query expression is the StoredQueryDescription (see Figure 6). A stored query description provides metadata describing the query, all of its user settable parameters, and the query expression itself.

All manipulations of stored queries (create, use, update, delete) are performed using the StoredQueryDescription. A stored query is referenced through the identifier (id attribute) of the stored query expression.

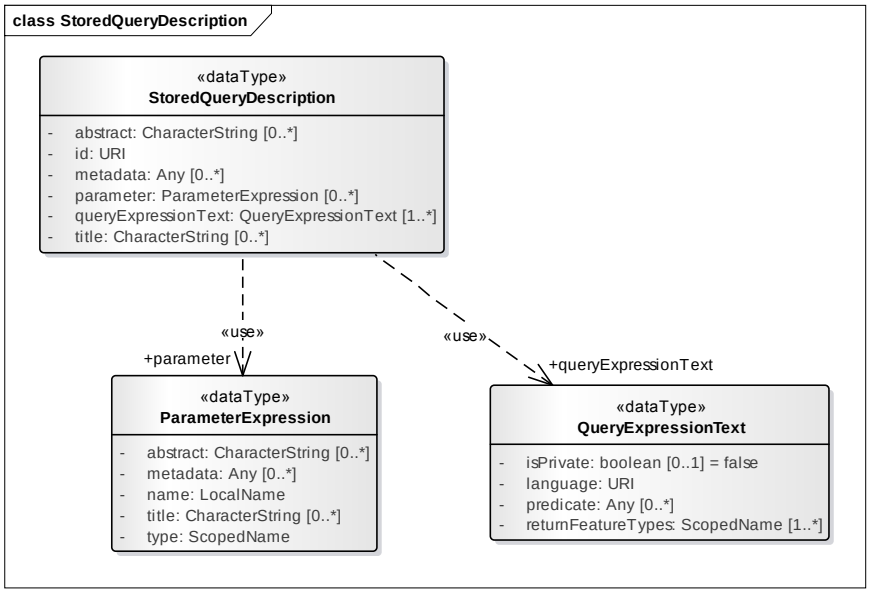

Figure 6 : Stored Query Description

The DGIWG Profile makes changes to the cardinality of some Stored Query Description elements. Those changes are described in Table 14. The NSG Profile makes no additional changes to this data structure.

Element Name		WFS Cardinality	DGIWG Cardinality	Description
Id		1..1	1..1	Unique identifier that can be used to repeatedly invoke a stored query
Title		0..*	1..*	Human-readable name to a stored query. One human readable title is mandatory.
Abstract		0..*	1..*	Descriptive human-readable to a stored query.
Metadata		0..*	0..*	Inline or reference more detailed metadata about a stored query (from OWS-Common)
Parameter[1]		0..*	0..*	Argument of a stored query
	Name	1..1	1..1	Name of each argument
	Type	1..1	1..1	Type of each argument
	Title	0..*	1..*	Human-readable name to an argument
	Abstract	0..*	1..*	Descriptive human-readable to an argument
	Metadata	0..*	0..*	Inline or reference more detailed metadata about an argument
QueryExpressionText		1..*	1..*	The wfs:QueryExpressionText element shall be used to enumerate one or more component query expressions that a stored query executes when invoked.
	ReturnFeatureTypes	1..*	1..*	Specifies the feature type(s) that each component query expression returns.
	Lanaguage	1..1	1..1	Specifies the implementation language of a component query expression. Services shall support the value "urn:ogc:def:queryLanguage:OGC-WFS::WFSQueryExpression". Optionally, the service may support languages such as SQL, XQuery, XPath, SPARQL, and Java.
	IsPrivate	0..1	0..1	Private means that the implementation text is visible only to the creator of the stored query.
	Predicate	0..*	0..*	A stored or ad-hoc query expression
[1] NOTE: when Parameter exists; Name, Type, Abstract, and Title are mandatory.				

Table 14 : DGIWG Stored Query Description

8 Supported Operations

The NSG WFS Profile implements the mandatory Conformance Classes of OGC 09-025r2 and the DGIWG Basic WFS Conformance Class (see Requirement 1). Those conformance classes require that the NSG WFS Profile implement the following operations:

- GetCapabilities,
- GetPropertyValue.
- GetFeature,
- DescribeFeatureType,

- ListStoredQueries,
- DescribeStoredQueries,

The NSG Profile's Locking WFS Conformance Class requires conformance to the DGIWG Locking WFS Conformance Class. Implementation of the following operations is required:

- LockFeature,
- GetFeatureWithLock,
- Transaction,

The NSG Profile's Manage Stored Queries Conformance Class requires conformance to the WFS Manage Stored Queries Conformance Class. Implementation of the following operations is required:

- CreateStoredQuery
- DropStoredQuery

The NSG Profile extends the DGIWG and OGC standards through the Enhanced Paging Conformance Class. That class adds the following operation:

- PageResults

For the most part operations are implemented in this Profile without change. Where modifications are required, the following sections describe the required changes and the reason for those changes.

8.1 GetCapabilities

The GetCapabilities operation generates a service metadata document describing a WFS service provided by a server. The service metadata document is modular. It consists of a core document which is an aggregation point for additional metadata documents. These additional metadata documents are called "sections". The OWS Common specification defines five of these sections:

- Service identification,
- Service provider,
- Content,
- Operations metadata, and
- IsoMetadata

The WFS standard adds the following sections to this set:

- WSDL – Web Service Description Language,
- FeatureTypes,
- FilterCapabilities.

A client can specify which of these sections they want included in the Service Metadata document when they issue the getCapabilities() request.

Requirement 20: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall demonstrate conformance to the requirements in Section 6 in their implementation of the GetCapabilities operation.

(modifies WFS 8 and DGIWG 7.2.1)

Rationale: This requirements serves to provide traceability between the Common Element requirements in Section 7 and the realization of those requirements by this operation.

8.1.1 Request

The GetCapabilities request is unchanged from its definition in the DGIWG Profile.

8.1.2 Response

The GetCapabilities response is unchanged from its definition in the DGIWG Profile.

8.1.3 Service Metadata

This NSG Profile does not add or delete any of the Service Metadata sections. However, some extensions are required. The following paragraphs discuss each section and what (if any) modifications are required for use in the NSG.

8.1.3.1 ServiceIdentification

The ServiceIdentification section of a service metadata document contains basic metadata about this specific service. NSG modifications to the DGIWG Profile come in two areas:

1) Modify element values to properly identify this service
2) Specify IC mandated security metadata

A UML model for ServiceIdentification section is provided in Figure 7. Each element of the Service Identification Metadata is then discussed in separate sections.

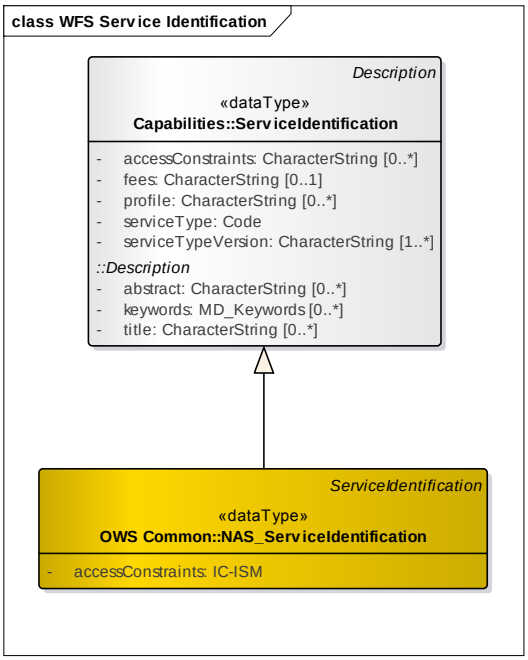

Figure 7 : Service Identification Metadata

8.1.3.1.1 Abstract

Requirement 21: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall clearly identify the profile used by the WFS service by including at least the following statement:

<Abstract>This service implements the NSG BASIC WFS profile of WFS 2.0</Abstract>.

(modifies DGIWG Requirement #2)

Requirement 22: A WFS Service that claims to be conformant to the NSG Locking WFS Conformance Class shall clearly identify the profile used by the WFS service by including at least the following statement:

<Abstract>This service implements the NSG LOCKING WFS profile of WFS 2.0</Abstract>.

(modifies DGIWG Requirement #16)

Rationale: This change distinguishes the NSG Profile from the DGIWG Profile and standard OGC implementations.

8.1.3.1.2 Title

This Profile imposes no additional requirements on the title element beyond those specified in the DGIWG Profile and OGC WFS standard.

8.1.3.1.3 Keywords

This Profile imposes no additional requirements on the keyword element beyond those specified in the DGIWG Profile.

8.1.3.1.4 ServiceTypeVersion

This Profile imposes no additional requirements on the ServiceTypeVersion element beyond those specified in the DGIWG Profile and OGC WFS standard.

8.1.3.1.5 ServiceType

This Profile imposes no additional requirements on the ServiceType element beyond those specified in the DGIWG Profile and OGC WFS standard.

8.1.3.1.6 Profile

Requirement 23: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall provide one of the following profile designations:

- <Profile>http://www.nga.mil/service/wfs/2.0/profile/basic</Profile>

- <Profile>urn:nga:service:wfs:2.0:profile:basic</Profile>

(modifies DGIWG Recommendation #2)

Requirement 24: A WFS Service that claims to be conformant to the NSG Locking WFS Conformance Class shall provide one of the following profile designations:

- <Profile>http://www.dgiwg.org/service/wfs/2.0/profile/locking</Profile>

- <Profile>urn:dgiwg:service:wfs:2.0:profile:locking</Profile>

(modifies DGIWG Recommendation #7)

Rationale: This change distinguishes the NSG Profile from the DGIWG Profile and standard OGC implementations.

8.1.3.1.7 Fees

This Profile imposes no additional requirements on the Fees element beyond those specified in the DGIWG Profile and OGC WFS standard.

8.1.3.1.8 AccessConstraints

NSG systems perform Identity and Access Management (IDAM) through an Attribute Based Access Control (ABAC) enterprise infrastructure. All requests to access a WFS service are authorized by a Policy Decision Point (PDP). PDP authorizations are granted according to the Security Policy governing the service. To make an access decision, the PDP must be supplied with the request, security markings of the service, and security attributes of the requestor. The NSG Profile supports this capability by specifying how to populate the security markings element of the Service Identification metadata.

Requirement 25: A WFS Service that claims to be conformant to the NSG Basic WFS Profile shall identify the highest classification level of the content accessible through the WFS service by populating the <ows:AccessConstraints>.

- The classification level shall be encoded using the most recent version of the IC Information Security Marking standard (IC.ISM)

- The classification level encoded in the IC.ISM markings shall be sufficiently restricted to dominate any content which may reside on the service.

(restricts DGIWG Requirement #4)

Rationale: The DGIWG Profile requires that all WFS services "shall identify the highest classification level of the content accessible through the WFS service". The mandated U.S. standard for classification markings is the IC.ISM. Therefore, those markings should be used to satisfy the DGIWG requirement. This requirement restricts the DGIWG requirement by mandating the IC.ISM.

8.1.3.2 ServiceProvider

The Service Provider document provides information about the organization and/or individuals responsible for this service.

This Profile imposes no additional requirements on the ServiceProvider document beyond those specified in the DGIWG Profile and OGC WFS standard.

8.1.3.3 OperationsMetadata

Operations Metadata provides information about the service operations offered by the service. The organization of the Operations Metadata is described in Figure 8. This metadata provides three types of information about the service:

1) Constraints which apply to the service as a whole,
2) Descriptions of each operation supported by the service,
3) Descriptions of parameters used by the operations.

Description of the parameters at this level is counter-intuitive. However, many parameters are used by more than one operation. They should have the same format, behavior, and constraints wherever they are used. Furthermore, some system constraints restrict the use and population of the parameters. Having all of this information at one place facilitates understanding and consistency.

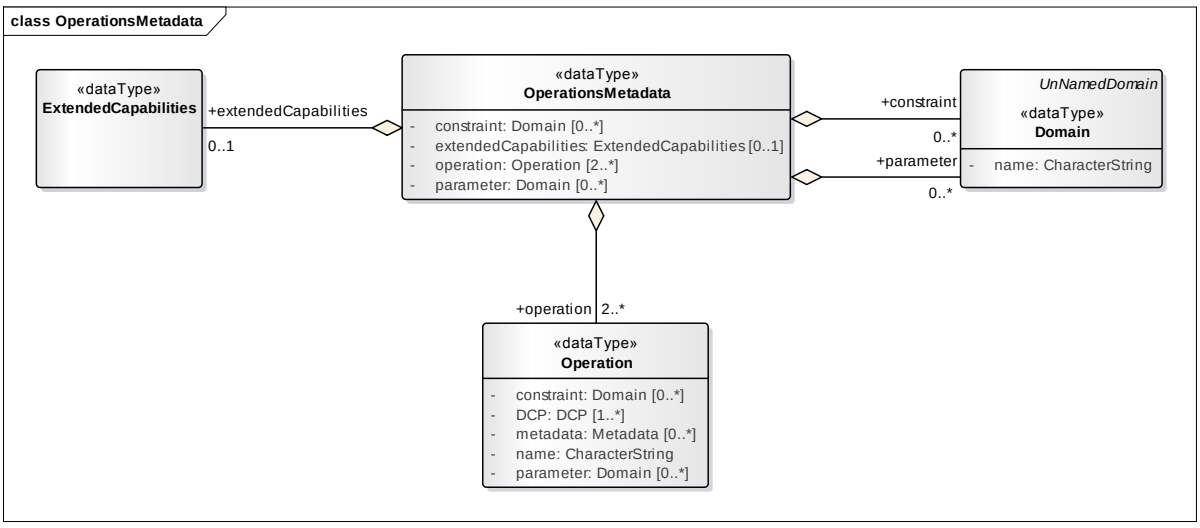

Figure 8 : WFS Operations Metadata

8.1.3.3.1 Service Constraints

The WFS Service Constraints are associated with the OperationsMetadata entity. They identify the presence or absense of a capability at the service level. This metadata is described in Table 15. The table has two sections. The upper section replicates the Service Constraints defined in WFS 2.0. Any changes to how those constraints should be applied are described in the description field. The lower section adds the constraints added by this Profile.

Requirement 26: A WFS Service that claims to be conformant to the NSG WFS Profile shall populate the Service Constraints of the OperationsMetadata document as described in Section 8.1.3.3.1.

(modifies WFS 8.3.5.3)

Rationale: The changes made to the WFS service by this Profile must be described in the Service Metadata. This requirement specifies how these changes shall be represented in the Constraints component of the OperationsMetadata document.

Constraint Name	Possible Values and/or Value Types	Description
WFS 2.0 Constraints		
ImplementsBasicWFS	Boolean value; either "TRUE" or "FALSE"	Indicates that the service implements the basic WFS conformance class.
ImpelmentsTransactionalWFS	Boolean value; either "TRUE" or "FALSE"	***MODIFIED: shall always have the same value as ImplementsLockingWFS***
ImplementsLockingWFS	Boolean value; either "TRUE" or "FALSE"	Indicates that the service implements the Locking conformance class
KVPEncoding	Boolean value; either "TRUE" or "FALSE"	***MODIFIED: shall always be TRUE***
XMLEncoding	Boolean value; either "TRUE" or "FALSE"	***MODIFIED: shall always be TRUE***
SOAPEncoding	Boolean value; either "TRUE" or "FALSE"	Indicates that the service implements the SOAP conformance class
ImplementsInheritance	Boolean value; either "TRUE" or "FALSE"	Indicates that the service implements the inheritance conformance class
ImplementsRemoteResolve	Boolean value; either "TRUE" or "FALSE	Indicates that the operation to which the constraint is specified can resolve references that are remote to the service. If the constraint is not specified then the value of FALSE shall be assumed.
ImplementsResultPaging	Boolean value; either "TRUE" or "FALSE"	***MODIFIED: shall always be TRUE***
ImplementsStandardJoins	Boolean value; either "TRUE" or "FALSE"	***MODIFIED: shall always be TRUE***
ImplementsSpatialJoins	Boolean value; either "TRUE" or "FALSE"	***MODIFIED: shall always be TRUE***
ImplementsTemporalJoins	Boolean value; either "TRUE" or "FALSE"	***MODIFIED: shall always be TRUE***
ImplementsFeatureVersioning	Boolean value; either "TRUE" or "FALSE"	***MODIFIED: shall always be TRUE***
ManageStoredQueries	Boolean value; either "TRUE" or "FALSE"	Indicates that the service implements the Manage Stored Queries conformance class.
NSG Profile Constraints		
ImplementsEnhancedPaging	Boolean value; either "TRUE" or "FALSE"	Indicates that the service implements the Enhanced Paging conformance class

Table 15 : WFS Service Constraints

8.1.3.3.2 Parameters

The Parameter element of the OperationsMetadata provides descriptions of the parameters used by operations supported by this service. Since some parameters are used by more than one operation, a single service-level definition simplifies the Service Metadata.

The parameter metadata is described in Table 16. The table has two sections. The upper section is a condensed form of the Parameters defined in WFS 2.0. Any changes to how those parameters should be described are described in the description field. The lower section adds the parameters added by this Profile.

Requirement 27: A WFS Service that claims to be conformant to the NSG WFS Profile shall populate the Parameters element of the OperationsMetadata document as described in Section 8.1.3.3.2.

(modifies WFS 8.3.5.3)

Rationale: The changes made to the WFS service by this Profile must be described in the Service Metadata. This requirement specifies how these changes shall be represented in the Parameters component of the OperationsMetadata document.

Operation Name	Parameter Name	Expected Value Type	Description / Possible Values
WFS 2.0 Constraints			
All operations (except GetCapabilities)	Version	String	Shall include the value "2.0.0". May include the value "1.1.0", "1.0.0" or other vendor specific version number
GetFeature, GetFeatureWithLock, Transaction	srsName	String URI or MIME type	List of CRS's that the WFS is capable of handling
GetCapabilities	acceptVersions	String	Shall include the value "2.0.0". May include the values "1.1.0", "1.0.0" or other vendor specific version.
GetCapabilities	acceptFormats	MIME type	Shall include the value "text/xml". May include other MIME types such as "text/html" or "text/plain" or any other vendor supported MIME type the service is capable of generating.
GetCapabilities	sections	String	Zero or more of the following list of values: "ServiceIdentification", "ServiceProvider", "OperationsMetadata", "FeatureTypeList", "Filter_Capabilities"
DescribeFeatureType, GetPropertyValue, GetFeature, PageResults, GetFeatureWithLock	outputFormat	String or MIME type	Shall include the value "application/gml+xml; version=3.2". May include any other string or MIME type that the service supports including previous version of GML.
GetPropertyValue, GetFeature, GetFeatureWithLock	resolve	String	Shall include the values “none” and “local”. May also include the values “remote” and “all” indicating that the service can resolve remote resource references.
CreateStoredQuery	language	anyURI	Shall include the value urn:ogc:def:queryLanguage:OGC-WFS::WFSQueryExpression. May also include other values indicating other languages are supported.
Transaction	inputFormat	String or MIME type	Shall include the value "application/gml+xml; version=3.2". May include any other string or

			MIME type that the service supports including previous version of GML.
Transaction	vendorId	string	Any string that is used as a vendor identifier for the wfs:Native element.
NSG Profile Constraints			
GetPropertyValue, GetFeature, PageResults, GetFeatureWithLock	Timeout	integer	Time in milliseconds after which this operation will terminate and return an error. If not provided or has a value of 0, then the timeout parameter is not supported.

Table 16 : WFS Parameters

8.1.3.3.3 Operation

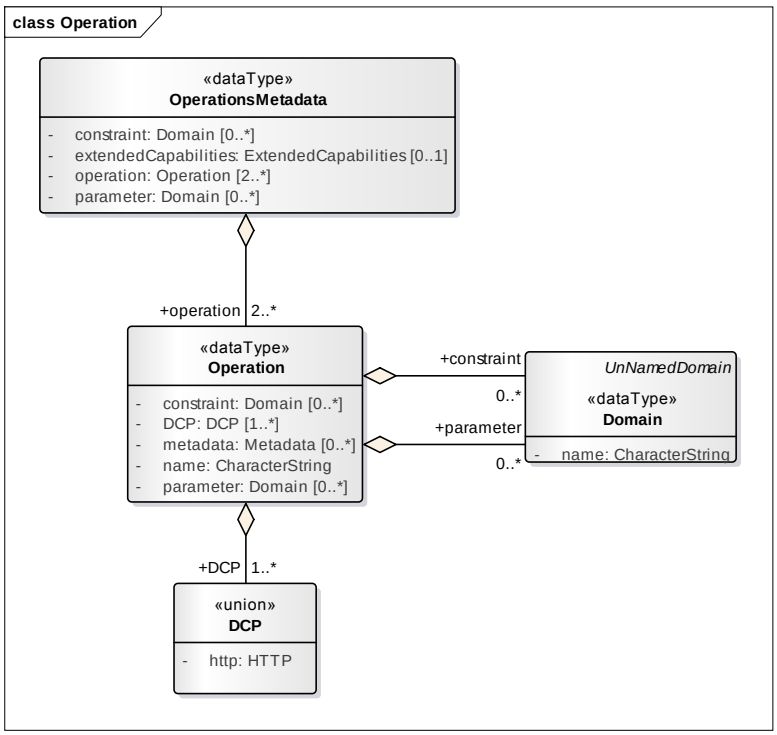

Figure 9 : Operation

The OperationsMetadata incudes an Operation element for each operation supported by the service. An Operation element provides information necessary to invoke that operation. That information includes:

- The name of the operation,
- Any additional parameter information beyond what is provided by the Operations Metadata (see Section 8.1.3.3.5),
- Any constraints on the operation (see Section 8.1.3.3.4),
- The Distributed Computing Platforms implementing this operation (see Section 8.1.3.3.6),
- Any additional descriptive metadata for this operation.

This Profile makes no additions to the operations defined by the OGC and DGIWG standards. The operations supported by the NSG Profile, organized by conformance class, are:

For the NSG Basic WFS Conformance Class:

- GetCapabilities,
- GetPropertyValue.
- GetFeature,
- DescribeFeatureType,
- ListStoredQueries,
- DescribeStoredQueries,

For the NSG Locking WFS Conformance Class:

- LockFeature,
- GetFeatureWithLock,
- Transaction,

For the NSG Manage Stored Queries Conformance Class:

- CreateStoredQuery
- DropStoredQuery

For the Enhanced Paging Conformance Class:

- PageResults

8.1.3.3.4 Operation Constraints

Most WFS operations have parameters who's values are restricted to a specific set of valid values. Operation Constraints document those valid values. A set of Operation Constraints is associated with each operation. Table 17 describes the operation constraints available to implementatsions of the OGC standards and the NSG Profile. Since a constraint is not applicable to all operations, the operations where it is applicable are listed as well.

Requirement 28: A WFS Service that claims to be conformant to the NSG WFS Profile shall populate the Operation Constraints of the Operation Metadata document as described in Section 8.1.3.3.4.

(modifies WFS 8.3.5.3)

Rationale: The changes made to the WFS service by this Profile must be described in the Operation Metadata. This requirement specifies how these changes shall be represented in the Constraints component of the Operation Metadata.

Constraint Name	Possible Values and/or Value Types	Default Value	WFS Operation	Description
WFS 2.0 Constraints				
AutomaticDataLocking	Boolean value; either "TRUE" or "FALSE"	FALSE	Transaction	***MODIFIED: should always be TRUE per DGIWG Profile***
PreservesSiblingOrder	Boolean value; either "TRUE" or "FALSE"	FALSE	Transaction	Specifies whether the service preserves sibling order for properties with cardinality greater than 1. If the value is true, the service shall preserve sibling order. Otherwise sibling order is not guaranteed to be preserved.
PagingIsTransactionSafe	Boolean value; either "TRUE" or "FALSE"	FALSE	GetFeature PageResults, GetFeatureWithLock GetPropertyValue	***MODIFIED: should always be FALSE per DGIWG Profile***

CountDefault	Integer value greater than or equal to zero.		GetFeature PageResults, GetFeatureWithLock GetPropertyValue	***MODIFIED: shall always be populated. A value of 10 is recommended.***
ResolveTimeoutDefault	Integer value greater than or equal to zero.		GetFeature GetFeatureWithLock GetPropertyValue	***MODIFIED: shall always be populated. A value of 300 is recommended.***
SortLevelLimit	Integer value greater than or equal to zero.		GetFeature GetFeatureWithLock	The SortLevelLimit constraint defines the maximum number of properties that may be simultaneously sorted. In the event that a request contains too many fes:SortProperty elements for a particular service (i.e. exceeds the SortLevelLimit constraint), the service shall respond with an exception. If the constraint is not specified then there is no limit to the number of sort properties that may be specified.
ResolveLocalScope	Integer greater than zero OR the character "*"	*	GetFeature GetFeatureWithLock GetPropertyValue	***MODIFIED: shall always be populated. A value of 3 is recommended.*** ***Note that this is contrary to the DGIWG Profile which specifies a range. WFS specifies an integer value.***
ResolveRemoteScope	Integer greater than zero OR the character "*"	*	GetFeature GetFeatureWithLock GetPropertyValue	***MODIFIED: shall always be populated. A value of 3 is recommended.*** ***Note that this is contrary to the DGIWG Profile which specifies a range. WFS specifies an integer value.***
ResponseCacheTimeout	Integer greater than zero		GetFeature GetFeatureWithLock GetPropertyValue	Define the length of time in seconds that responses shall be cached to support paging. If the constraint is not specified then the response cache never times out.
QueryExpressions	QName; one of wfs:Query or qfs:StoredQuery		GetFeature GetFeatureWithLock GetPropertyValue LockFeature	The names of the supported query expression elements.
NSG Profile Constraints				
Authentication	URI		All	This URI identifies the Identification and Authentication method used by the operation(s). The URI should be resolvable to an on-line resource which provides a detailed description of the Authentication method.
RequestTimeoutDefault	Integer greater than zero	300	GetFeature PageResults, GetFeatureWithLock GetPropertyValue	The default value for the timeout parameter of the Standard Presentation Parameters. If not provided or has a value of 0, then the timeout parameter is not supported.

Table 17 : WFS Operation Constraints

8.1.3.3.5 Operation Parameters

The Parameter element of the OperationsMetadata provides descriptions of the parameters used by operations supported by this service. However, in some cases the use of a parameter for a specific operation may differ from the service-level description. The parameters element on the Operation element allows an operation to override the service level definition.

The operation parameters are described in Table 18. The table has two sections. The upper section is a extracted from the Parameters defined in WFS 2.0. Any changes to how those parameters should be described are described in the description field. The lower section adds the parameters added by this Profile. Note that each description applies to one and only one operation.

Requirement 29: A WFS Service that claims to be conformant to the NSG WFS Profile shall populate the Parameters element of the Operation element as described in Section 8.1.3.3.5.

(modifies WFS 8.3.5.3)

Rationale: The changes made to the WFS service by this Profile must be described in the Service Metadata. This requirement specifies how these changes shall be represented in the Parameters component of the Operation elements of the Operations Metadata document.

Operation Name	Parameter Name	Expected Value Type	Description / Possible Values
WFS 2.0 Constraints			
DescribeFeatureType,	outputFormat	String or MIME type	**MODIFIED[1]: Shall include the value "application/gml+xml; version=3.2".**
NSG Profile Constraints			
None defined			
[1] This change restricts the outputFormat parameter as used by the DescribeFeatureType operation to an enumerated list of MIME types for schema languages.			

Table 18 : Operation Parameters

8.1.3.3.6 DCPs

The DCP element defines the protocols and services used by the service to interoperate with its clients. At this time, all of the OGC bindings are described through the HTTP DCP. This Profile makes no additional requirements on the DCP element beyond those specified in the OGC standard.

8.1.3.4 Content

The Content document provides information about the content provided by this service. Content metadata is not supported by the OGC WFS standard, the DGIWG Profile, nor by the NSG Profile.

8.1.3.5 WSDL

This Profile imposes no additional requirements on the WSDL element beyond those specified in the DGIWG Profile and OGC WFS standard.

8.1.3.6 FeatureTypes

This Profile imposes no additional requirements on the FeatureTypes element beyond those specified in the DGIWG Profile and OGC WFS standard. Implementers should be aware

that the DGIWG Profile does modify the FeatureType class from the definition in the OGC WFS standard. Those changes are summarized in Table 19.

Element Name			WFS Cadinality	DGIWG Cardinality	Description
Name			1..1	1..1	Name of the feature type
Title			0..*	1..*	One human readable title is mandatory for all cases.
Abstract			0..*	1..*	One abstract element is recommended for all cases.
keywords			0..*	1..*	At least one keyword is required for each feature type to facilitate data discovery in catalogues. Keywords can be duplicated in multiple languages. Using DFDD codes and names as keywords are recommended.
Choice	NoCRS		1..1	1..1	The NoCRS element shall be used for feature types that have no spatial properties. If NoCRS is not provided, then SupportedCRS is required.
	SupportedCRS				
		DefaultCRS	1..1	1..1	The DefaultCRS element indicates which coordinate reference system shall be used by a WFS.
		OtherCRS	0..*	1..*	Additional coordinate reference systems
OutputFormats (list)			0..*	0..*	If this optional element is not specified, then all the result formats listed for the GetFeature operation are assumed to be supported.
Envelope			0..*	1..*	This knowledge aids client applications by letting them know where they should query in order to have a high probability of finding feature data.
Metadata			0..*	1..*	When MetadataURL is used, the xlink:href element shall be used to reference any metadata. In an SDI like architecture, a link to the metadata resource available on a CSW service should be provided.
ExtendedDescription			0..1	0..1	A WFS may add elements to the description of a feature type, without having to redefine the capabilities schema, using the ExtendedDescription element.

Table 19 : Elements to Describe Feature Types

8.1.3.7 FilterCapabilities

This Profile imposes no additional requirements on the FilterCapabilities element beyond those specified in the DGIWG Profile and OGC WFS standard.

8.2 DescribeFeatureType

Requirement 30: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall demonstrate conformance to the requirements in Sections 7.1, 7.2, and 7.5 in their implementation of the DescribeFeatureType operation.

(modifies WFS 10 and DGIWG 7.2.4)

Rationale: This requirements serves to provide traceability between the Common Element requirements in Section 7 and the realization of those requirements by this operation.

8.3 GetPropertyValue

Requirement 31: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall demonstrate conformance to the requirements in Sections 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7, 7.8.1, and 7.8.2 in their implementation of the GetPropertyValue operation.

(modifies WFS 10 and DGIWG 7.2.4)

Rationale: This requirements serves to provide traceability between the Common Element requirements in Section 7 and the realization of those requirements by this operation.

8.4 GetFeature

Requirement 32: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall demonstrate conformance to the requirements in Sections 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7, 7.8.1, and 7.8.2 in their implementation of the GetFeature operation.

(modifies WFS 11 and DGIWG 7.2.3)

Rationale: This requirements serves to provide traceability between the Common Element requirements in Section 7 and the realization of those requirements by this operation.

8.5 PageResults

Requirement 33: A WFS Service that claims to be conformant to the NSG Enhanced Paging Conformance Class shall implement the PageResults operation as described in Section 8.5.

(extends DGIWG WFS Profile)

Rationale: Random access to result set pages is commonly supported by discovery and retrieval standards including the ODNI CDR suite. The OGC WFS standard only supports sequential navigation of a result set through the “next” and “previous” parameters. While it is possible to mimic random access using a sequential access capability, it is far from an optional solution. This operation extends the WFS paging model to make random access an integral part of the WFS service.

Requirement 34: A WFS Service that claims to be conformant to the NSG Enhanced Paging Conformance Class shall demonstrate conformance to the requirements in Sections 7.1, 7.2, 7.3, 7.5, 7.6, and 7.7 in their implementation of the PageResults operation.

(extends DGIWG WFS Profile)

Rationale: The PageResults operation is essentially a GetFeatures operation where the query expression has been replaced by the identifier for the previously generated result set (see Figure 10).

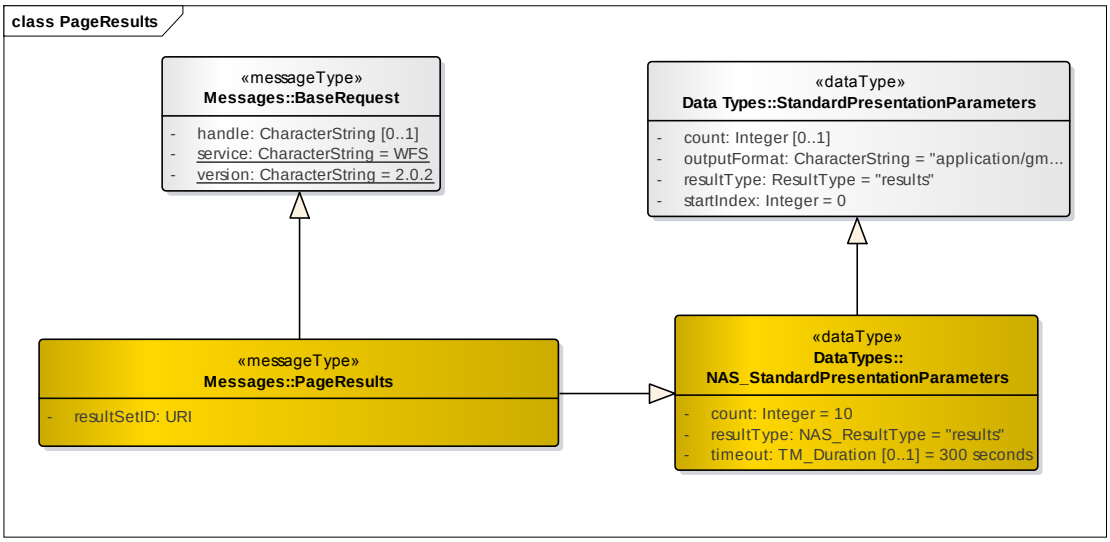

Figure 10 : PageResults Operation

8.6 LockFeature

Requirement 35: A WFS Service that claims to be conformant to the NSG Locking WFS Conformance Class shall demonstrate conformance to the requirements in Sections 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7, 7.8.1, and 7.8.2 in their implementation of the LockFeature operation.

(modifies WFS 12 and DGIWG 8.2.3)

Rationale: This requirements serves to provide traceability between the Common Element requirements in Section 7 and the realization of those requirements by this operation.

8.7 GetFeatureWithLock

Requirement 36: A WFS Service that claims to be conformant to the NSG Locking WFS Conformance Class shall demonstrate conformance to the requirements in Sections 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7, 7.8.1, and 7.8.2 in their implementation of the GetFeatureWithLock operation.

(modifies WFS 13 and DGIWG 8.2.3)

Rationale: This requirements serves to provide traceability between the Common Element requirements in Section 7 and the realization of those requirements by this operation.

8.8 Transaction

Requirement 37: A WFS Service that claims to be conformant to the NSG Locking WFS Conformance Class shall demonstrate conformance to the requirements in Sections 7.1, 7.2, 7.3, 7.4, 7.5, 7.8.1, and 7.8.2 in their implementation of the Transaction operation.

(modifies WFS 5 and DGIWG 8.2.2)

Rationale: This requirements serves to provide traceability between the Common Element requirements in Section 7 and the realization of those requirements by this operation.

8.9 ListStoredQueries

Requirement 38: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall demonstrate conformance to the requirements in Sections 7.2 and 7.5 in their implementation of the ListStoredQueries operation.

(modifies WFS 14.3 and DGIWG 7.2.5)

Rationale: This requirements serves to provide traceability between the Common Element requirements in Section 7 and the realization of those requirements by this operation.

8.10 DescribeStoredQueries

Requirement 39: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall demonstrate conformance to the requirements in Sections 7.2, 7.3, 7.4, 7.5, 7.8.1, and 7.8.2 in their implementation of the DescribeStoredQueries operation.

(modifies WFS 14.4 and DGIWG 7.2.5)

Rationale: This requirements serves to provide traceability between the Common Element requirements in Section 7 and the realization of those requirements by this operation.

8.11 DropStoredQuery

Requirement 40: A WFS Service that claims to be conformant to the NSG Manage Stored Queries Conformance Class shall demonstrate conformance to the requirements in Sections 7.2 and 7.5 in their implementation of the DropStoredQuery operation.

(modifies WFS 14.6)

Rationale: This requirements serves to provide traceability between the Common Element requirements in Section 7 and the realization of those requirements by this operation.

8.12 CreateStoredQuery

Requirement 41: A WFS Service that claims to be conformant to the NSG Manage Stored Queries Conformance Class shall demonstrate conformance to the requirements in Sections 7.1, 7.2, 7.3, 7.4, 7.5, 7.8.1, and 7.8.2 in their implementation of the CreateStoredQuery operation.

(modifies WFS 14.5)

Rationale: This requirements serves to provide traceability between the Common Element requirements in Section 7 and the realization of those requirements by this operation.

9 Additional Material

This section contains information which is referenced from previous sections of this standard. It is being documented separately due to its size and complexity.

9.1 Resource Identifiers

NSG requirements described in this specification, as well as NSG strategies for content management, impose requirements on resource identifiers and their behavior. This section

addresses how those requirements will be addressed when a Resource Identifier is used with a WFS.

9.1.1 Description:

The NSG Profile of the WFS shall recognize two types of resource identifier. The Instance Resource Identifier shall serve as a universally unique identifier for a specific instance of data. The Entity Resource Identifier shall serve as an identifier for the concept which the data represents.

Instance Identifier:

a) Only identifier which shall be used with the GetResourceByID() operation.

b) A resource shall have one and only one Instance Identifier.

c) Two resources with the same Instance Identifier shall have identical content.

d) Instance Identifiers shall be either a UUID or a GUIDE ID.

In a GML document, the Instance Identifier shall be captured in the identifier property of AbstractGML (see Figure 11).

Entity Identifier:

a) Multiple versions of a resource shall all have the same Entity Identifier.

b) A resource may have more than one Entity Identifier.

c) Entity identifiers shall be assigned by the organization with custodial responsibility for that resource.

d) The default form for an Entity Identifier is the UUID.

e) An Entity Identifier shall NOT be a GUIDE ID.

In a GML document, the Entity Identifier shall be captured in the name property of AbstractGML (see Figure 11).

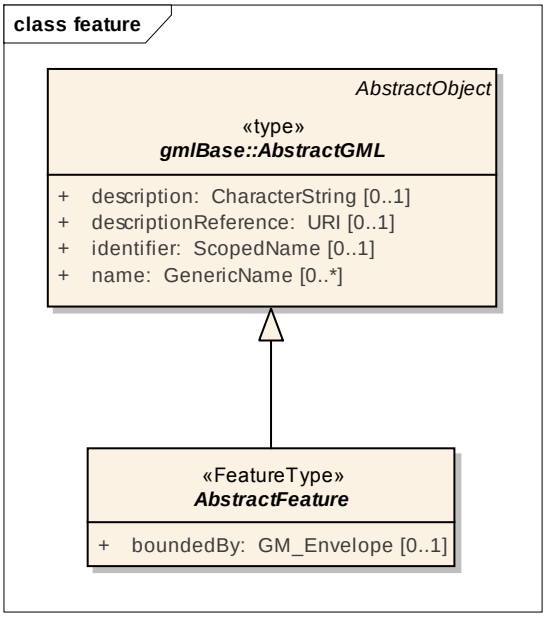

Figure 11 : GML Feature Class

9.1.2 URIs In the NSG

Uniform Resource Identifiers (URI) are a standard way to identify resources in a Web environment. The most common form of URI is the Uniform Resource Locator (URL). However, the URL is commonly interpreted as a locator, a path that resolves to the desired resource. This works well in an integrated environment such as the World Wide Web. The NSG, however, is an environment where there are multiple security levels, enclaves, guards, disconnected and semi-disconnected systems, and highly mobile systems. More often than not a URL will not resolve to a resource.

The other form of URI is the Uniform Resource Name (URN). The URN is a global unique identifier for a resource. However it is not a locator for that resource. It is up to a resolver service to convert the URN into a path to the resource. In the World Wide Web this is an unnecessary complexity. For the IC and DoD, however, it is essential. The valid path to a resource will be different in each enclave. A system needs to know the local path to the resource if it is going to retrieve is successfully. URNs are appropriate for that use.

Therefore, URLs should not be used as identifiers within the NSG except in those cases where there is a high level of confidence that they will always resolve.

While the URI standard provides direction on how to format an identifier, it does not describe how to populate the URI.

The Universally Unique Identifier (UUID) is a scheme for globally unique persistent identifiers which has been pervasive in IT systems for over 30 years. It can be stored as a 128 bit binary value or as a 32 character hexadecimal representation of that value. While it has more than proven its worth, it is not a URI. Therefore, the UUID should only be used where field-length or format limitations make a longer identifier impractical.

IETF RFC 4122 defines a URN encoding of the UUID. Where there is sufficient space and the GUIDE id is not practical, the URN encoding of the UUID may be used. An example of the URN form of a UUID is:

urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6

The mandated resource identifier within the U.S. Intelligence Community is the GUIDE id. The format of the GUIDE id is as follows:

guide://{prefix}/{suffix}

Prefix values are allocated in a block to each IC agency. For NGA systems, prefixes are allocated to systems which have custodial responsibility for content.

Suffix values are defined by each IC agency. For NGA systems, the suffix is the hexadecimal representation of the UUID.

Identifier	Use
GUIDE ID	Whenever possible
URN form of UUID	When there is sufficient room for a URN but no prefix is available
Native form of UUID	When there is insufficient room for the URN form

Table 20 : Identifier Usage Guidance

9.2 Resource Versioning

9.2.1 WFS Resource Versioning Support

The WFS 2.0 Standard states that:

"If the service supports versioning of features, then the service shall maintain version information about each feature instance. This International Standard makes no assumptions about how that version information is maintained. Functions defined in the Filter Encoding Standard (see ISO 19143, 7.11.2) allow version navigation based on the resource identifier."

For WFS Services conformant to the NSG Basic WFS Conformance Class, this clause shall read:

"The service shall maintain version information about each feature instance. At a minimum this version information shall consist of a resource identifier and the date and time upon which this version was first produced or last updated. The version information shall be maintained so as to allow version navigation as defined in the Filter Encoding Standard (see ISO 19143, 7.11.2)."

There are three main points to this clause:

1) Feature versioning support is mandated
2) Feature versioning shall be based on date and time
3) Feature versioning shall be compatible with the Filter Encoding standard

Feature versioning is mandated to provide consistent behavior across all NSG WFS implementations. A service which hosts content which is not versioned shall utilize the default behavior described in Table 21.

It is desirable that there be one versioning construct for all of the NSG. Due to the velocity of SOM and ABI data, the time stamp of the last update is the most versatile approach.

The Feature Encoding Standard provides a mechanism to navigate through versions of Features. This mechanism shall be leveraged to minimize impact to systems and interoperability.

9.2.2 Filter Encoding

The OGC Filter Encoding standard defines the ResourceID query structure as shown in Figure 12

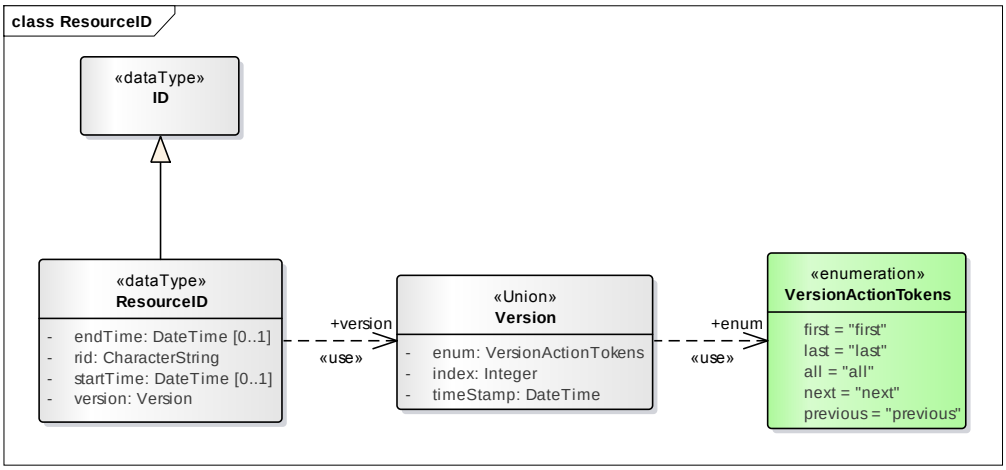

Figure 12 : OGC Filter Encoding Resource IDs

The ResourceID operator includes the following elements:

1) Rid -> the Entity Resource Identifier (see Section 9.1)
2) StartTime and endTime -> provides a temporal bound for the versions desired
3) Version -> the version navigation operation to execute

The attributes startTime and endTime may be used to select all versions of a resource between the specified start date and end date. The startTime and endTime attributes shall always be specified together. If the startTime and endTime are specified, the version attribute shall not be specified.

The version attribute may be an index (accessing the version of the resource), a timestamp (accessing the version of the resource closest to the specified date) or a token (“first“, “latest“, “previous“, “next“, and “all“). The behavior of the service upon receipt of these values shall behave as described in Table 21.

TimeStamp	
	Pre-conditions: none
	Variables and Functions: <timestamp> == time value passed in operation <rid> == rid parameter from ResourceID <eid> == entity id from a resource <x> == resource to return, initialized to null <dtr> == difference between returned resource time and timestamp <rset> == set of resources sorted by timestamp, initialized to null

	Behavior:
	DO FOR each resource <y> where <eid> = <rid>
	Insert_sorted(<y>, <rset>)
	ENDDO
	DO FOR each resource <y> in <rset>
	IF <x> = null THEN
	<x> == <y>
	<dtr> == timestamp(<x>) – <timestamp>
	ELSE

	IF |(timestamp(<y>) - <timestamp>)| < |<dtr>| THEN
	<x> == <y>
	<dtr> == timestamp(<y>) - <timestamp>
	ENDIF
	ENDIF
	ENDDO
	RETURN <x>

	Default Behavior: RETURN null
	Exception Codes: InvalidParameterValue
	Notes: This code leaves a sorted result set which can then be navigated using next, first, etc.
Index	
	Pre-conditions: none
	Variables and Functions: <index> == index value passed in operation <rid> == rid parameter from ResourceID <eid> == entity id from a resource <x> == resource to return, initialized to null <rset> == set of resources sorted by timestamp

	Behavior:
	IF <rset> = null THEN
	DO FOR each resource <y> where <eid> = <rid>
	Insert_sorted(<y>, <rset>)
	ENDDO
	ENDIF
	<x> == extract(<index>, <rset>)
	RETURN <x>

	Default Behavior: RETURN null
	Exception Codes: InvalidParameterValue
First	
	Pre-conditions: none
	Variables and Functions: <timestamp> == time value passed in operation <rid> == rid parameter from ResourceID <eid> == entity id from a resource <x> == resource to return, initialized to null <dtr> == difference between returned resource time and timestamp

	Behavior:
	IF <rset> = null THEN
	DO FOR each resource <y> where <eid> = <rid>
	Insert_sorted(<y>, <rset>)
	ENDDO
	ENDIF

	<x> == extract(0, <rset>) RETURN <x>
	Default Behavior: RETURN null
	Exception Codes: InvalidParameterValue
Last	
	Pre-conditions: none
	Variables and Functions: <index> == index of the last returned resource <rid> == rid parameter from ResourceID <eid> == entity id from a resource <x> == resource to return, initialized to null <rset> == sorted resource set to return

	Behavior:
	IF <rset> = null THEN
	DO FOR each resource <x> where <eid> = <rid>
	Insert_sorted(<x>, <rset>)
	ENDDO
	ENDIF
	<index> == 0
	DO FOR each resource <x> in <rset>
	<index> == <index> + 1
	ENDDO
	<index> == <index> - 1
	<x> == extract(<index>, <rset>)
	RETURN <x>

	Default Behavior: RETURN null
	Exception Codes: InvalidParameterValue
All	
	Pre-conditions: none
	Variables and Functions: <rid> == rid parameter from ResourceID <eid> == entity id from a resource <rset> == resource set to return, initialized to null

	Behavior:
	IF <rset> = null THEN
	DO FOR each resource <y> where <eid> = <rid>
	Insert_sorted(<y>, <rset>)
	ENDDO
	ENDIF
	RETURN <rset>

	Default Behavior: RETURN null
	Exception Codes: InvalidParameterValue
Next	
	Pre-conditions: none
	Variables and Functions: <index> == index of the last returned resource <rid> == rid parameter from ResourceID <eid> == entity id from a resource <x> == resource to return, initialized to null <rset> == resource set to return, initialized to null

	Behavior:
	IF <rset> = null THEN
	DO FOR each resource <x> where <eid> = <rid>
	Insert_sorted(<x>, <rset>)
	ENDDO
	<index> == -1
	ENDIF
	<index> == <index> + 1
	<x> == extract(<index>, <rset>)
	RETURN <x>

	Default Behavior: RETURN null
	Exception Codes: InvalidParameterValue
Previous	
	Pre-conditions: none
	Variables and Functions: <index> == index of the last returned resource <rid> == rid parameter from ResourceID <eid> == entity id from a resource <x> == resource to return, initialized to null <rset> == resource set to return, initialized to null

	Behavior:
	IF <rset> = null THEN
	<index> == 0
	DO FOR each resource <x> where <eid> = <rid>
	Insert_sorted(<x>, <rset>)
	<index> == <index> + 1
	ENDDO
	ENDIF
	<index> == <index> - 1
	<x> == extract(<index>, <rset>)
	RETURN <x>

	Default Behavior: RETURN null
	Exception Codes: InvalidParameterValue

Table 21 : Version Operators

9.2.3 Content

Two information elements are required on each resource if the OGC Filter versioning operations are to work. These are the Entity Resource Identifier and the Currency Timestamp. The Entity Resource Identifier is defined in section 9.1. For the time stamp, implementers may use a property defined within their resource model if an appropriate property exists. Failing that, implementations should use the EnvelopeWithTimePeriod subclass of GM_Envelope as shown in Figure 13. This class provides both a spatial and temporal extent for the resource. If the resource represents the state at a single point in time, the beginPosition and endPosition should both have the same value.

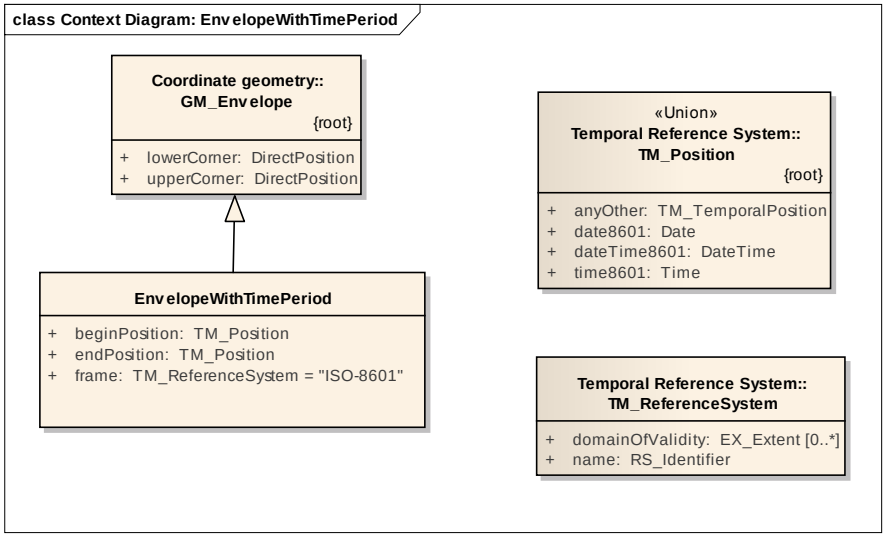

Figure 13 : Envelope With Time Period

Annex A Abstract Test Suite

Table 22, Requirement Test Matrix, lists the conformance requirements from sections 6, 7, and 8; and identifies the applicable abstract tests in sections A.1 through A.10. The abstract tests specify the methods to verify that an Implementation Under Test (IUT) conforms to the stated requirements.

Requirements
Requirement 1: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall demonstrate conformance to the DGIWG Basic WFS Conformance Class.
Tests: A.1.1,
Requirement 2: A WFS Service that claims to be conformant to the NSG Locking WFS Conformance Class shall demonstrate conformance to the NSG Basic WFS and DGIWG Locking WFS Conformance Classes.
Tests: A.1.2
Requirement 3: A WFS Service that claims to be conformant to the NSG Manage Stored Queries Conformance Class shall demonstrate conformance to the NSG Basic WFS Conformance Class and with the Manage Stored Queries Conformance Class from the OGC WFS standard.
Tests: A.1.3
Requirement 4: A WFS Service that claims to be conformant to the NSG Inheritance WFS Conformance Class shall demonstrate conformance to the NSG Basic WFS Conformance Class and with the Inheritance Conformance Class from the OGC WFS standard.
Tests: A.1.4
Requirement 5: A WFS Service that claims to be conformant to the NSG Remote Resolve WFS Conformance Class shall demonstrate conformance to the NSG Basic WFS Conformance Class and with the Remote Resolve Conformance Class from the OGC WFS standard.
Tests: A.1.5
Requirement 6: A WFS Service that claims to be conformant to the NSG SOAP WFS Conformance Class shall demonstrate conformance to the NSG Basic WFS Conformance Class and with the SOAP Conformance Class from the OGC WFS standard.
Tests: A.2.1, A.2.2, A.2.3, A.2.4, A.2.5, A.2.6, A.2.7, A.2.8, A.2.9, A.2.10, A.2.11, A.2.12
Requirement 7: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall implement service bindings as described in section 6.5. ***(expands on WFS 6.4, DGIWG 7.5, and DGIWG 8.4)***
Tests: A.2.1, A.2.2, A.2.3, A.2.4, A.2.5, A.2.6, A.2.7, A.2.8, A.2.9, A.2.10, A.2.11, A.2.12
Requirement 8: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall encode Features as described in section 7.1. ***(expands on WFS 7.1 and DGIWG 6.5)***
Test: A.3.1
Requirement 9: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall implement resource identifiers as described in section 9.1 ***(expands on WFS 7.2.1 and 7.2.2)***
Test: A.3.2

Requirements
Requirement 10: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall support Resource Versioning as described in section 9.2. This requirement is a restriction on Resource Versioning as described in OGC WFS 2.0 and OGC Feature Encoding 2.0. ***(replaces WFS 7.2.3 and restricts DGIWG 6.4)***
Test: A.3.3
Requirement 11: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall implement Predicate Languages as described in section 7.4. ***(expands on WFS 7.4, clarifies DGIWG 7.2.5.1)***
Test: A.3.4
Requirement 12: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall implement Exception Reporting as described in section 7.5. ***(expands on WFS 7.5)***
Test: A.3.5
Requirement 13: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall support the "count" Standard Presentation Parameter. ***(modifies WFS 7.6.3)***
Test: A.4.2
Requirement 14: The default value of the "count" Standard Presentation Parameter shall be 10. ***(modifies WFS 7.6.3)***
Test: A.4.2
Requirement 15: A WFS Service that claims to be conformant to the NSG Basic WFS Profile may implement the "timeout" extension to the Standard Presentation Parameters. If implemented, this parameter shall comply with the following requirements: ***- The timeout parameter shall specify the maximum clock time the WFS should spend processing the request.*** ***- The units of the timeout parameter shall be seconds.*** ***- Upon expiration of the specified timeout period, the WFS shall terminate the request.*** ***- Upon expiration of the specified timeout period, the WFS shall return an OperationProcessingTimeout error to the client.*** ***(extends WFS 7.6.3)***
Test: A.4.5
Requirement 16: A WFS Service that claims to be conformant to the NSG Basic WFS Profile shall support the mandatory formats and parameter values as described in Table 7 for the outputFormat parameter of the StandardPresentationParameters. This support shall be in addition to the formats requirement in WFS 7.6.3.7 ***(modifies WFS 7.6.3)***
Test: A.4.4
Requirement 17: A WFS Service that claims to be conformant to the NSG Basic WFS Profile may support an additional ResultType value of "index". If supported, the index result type shall be implemented as described in Section 7.6.4

Requirements
(modifies WFS 7.6.3.1)
Tests: A.4.3, A.7.2
Requirement 18: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall implement the paging operations described in WFS Section 7.7.4.4.   ***(restricts WFS 2 and DGIWG Recommendation 6)***
Tests: A.7.5, A.7.2
Requirement 19: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class and supports the "index" result type, shall include the resultSetID element in StandardResponseParameters.   ***(extends WFS 7.7)***
Test: A.7.4
Requirement 20: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall demonstrate conformance to the requirements in Section 6 in their implementation of the GetCapabilities operation.   ***(modifies WFS 8 and DGIWG 7.2.1)***
Test: A.2.1
Requirement 21: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall clearly identify the profile used by the WFS service by including at least the following statement: ***<Abstract>This service implements the NSG BASIC WFS profile of WFS 2.0</Abstract>.***   ***(modifies DGIWG Requirement #2)***
Test: A.8.2.1
Requirement 22: A WFS Service that claims to be conformant to the NSG Locking WFS Conformance Class shall clearly identify the profile used by the WFS service by including at least the following statement: ***<Abstract>This service implements the NSG LOCKING WFS profile of WFS 2.0</Abstract>.***   ***(modifies DGIWG Requirement #16)***
Test: A.8.2.1
Requirement 23: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall provide one of the following profile designations: ***- <Profile>http://www.nga.mil/service/wfs/2.0/profile/basic</Profile>***   ***- <Profile>urn:nga:service:wfs:2.0:profile:basic</Profile>***   ***(modifies DGIWG Recommendation #2)***
Test: A.8.2.2
Requirement 24: A WFS Service that claims to be conformant to the NSG Locking WFS Conformance Class shall provide one of the following profile designations: ***- <Profile>http://www.dgiwg.org/service/wfs/2.0/profile/locking</Profile>***

Requirements
- <Profile>urn:dgiwg:service:wfs:2.0:profile:locking</Profile> ***(modifies DGIWG Recommendation #7)***
Test: A.8.2.2
Requirement 25: A WFS Service that claims to be conformant to the NSG Basic WFS Profile shall identify the highest classification level of the content accessible through the WFS service by populating the <ows:AccessConstraints>. ***- The classification level shall be encoded using the most recent version of the IC Information Security Marking standard (IC.ISM)*** ***- The classification level encoded in the IC.ISM markings shall be sufficiently restricted to dominate any content which may reside on the service.*** ***(restricts DGIWG Requirement #4)***
Test: A.8.2.3
Requirement 26: A WFS Service that claims to be conformant to the NSG WFS Profile shall populate the Service Constraints of the OperationsMetadata document as described in Section 8.1.3.3.1. ***(modifies WFS 8.3.5.3)***
Test: A.8.3.2
Requirement 27: A WFS Service that claims to be conformant to the NSG WFS Profile shall populate the Parameters element of the OperationsMetadata document as described in Section 8.1.3.3.2. ***(modifies WFS 8.3.5.3)***
Test: A.8.3.4
Requirement 28: A WFS Service that claims to be conformant to the NSG WFS Profile shall populate the Operation Constraints of the Operation Metadata document as described in Section 8.1.3.3.4. ***(modifies WFS 8.3.5.3)***
Test: A.8.3.3
Requirement 29: A WFS Service that claims to be conformant to the NSG WFS Profile shall populate the Parameters element of the Operation element as described in Section 8.1.3.3.5. ***(modifies WFS 8.3.5.3)***
Test: A.8.3.4
Requirement 30: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall demonstrate conformance to the requirements in Sections 7.1, 7.2, and 7.5 in their implementation of the DescribeFeatureType operation. ***(modifies WFS 10 and DGIWG 7.2.4)***
Test: A.2.2
Requirement 31: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall demonstrate conformance to the requirements in Sections 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7, 7.8.1, and 7.8.2 in their implementation of the GetPropertyValue operation. ***(modifies WFS 10 and DGIWG 7.2.4)***

Requirements
Test: A.2.3
Requirement 32: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall demonstrate conformance to the requirements in Sections 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7, 7.8.1, and 7.8.2 in their implementation of the GetFeature operation. ***(modifies WFS 11 and DGIWG 7.2.3)***
Test: A.2.4
Requirement 33: A WFS Service that claims to be conformant to the NSG Enhanced Paging Conformance Class shall implement the PageResults operation as described in Section 8.5. ***(extends DGIWG WFS Profile)***
Test: A.2.12
Requirement 34: A WFS Service that claims to be conformant to the NSG Enhanced Paging Conformance Class shall demonstrate conformance to the requirements in Sections 7.1, 7.2, 7.3, 7.5, 7.6, and 7.7 in their implementation of the PageResults operation. ***(extends DGIWG WFS Profile)***
Test: A.2.12
Requirement 35: A WFS Service that claims to be conformant to the NSG Locking WFS Conformance Class shall demonstrate conformance to the requirements in Sections 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7, 7.8.1, and 7.8.2 in their implementation of the LockFeature operation. ***(modifies WFS 12 and DGIWG 8.2.3)***
Test: A.2.7
Requirement 36: A WFS Service that claims to be conformant to the NSG Locking WFS Conformance Class shall demonstrate conformance to the requirements in Sections 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7, 7.8.1, and 7.8.2 in their implementation of the GetFeatureWithLock operation. ***(modifies WFS 13 and DGIWG 8.2.3)***
Test: A.2.8
Requirement 37: A WFS Service that claims to be conformant to the NSG Locking WFS Conformance Class shall demonstrate conformance to the requirements in Sections 7.1, 7.2, 7.3, 7.4, 7.5, 7.8.1, and 7.8.2 in their implementation of the Transaction operation. ***(modifies WFS 5 and DGIWG 8.2.2)***
Test: A.2.9
Requirement 38: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall demonstrate conformance to the requirements in Sections 7.2 and 7.5 in their implementation of the ListStoredQueries operation. ***(modifies WFS 14.3 and DGIWG 7.2.5)***
Test: A.2.5
Requirement 39: A WFS Service that claims to be conformant to the NSG Basic WFS Conformance Class shall demonstrate conformance to the requirements in Sections 7.2, 7.3, 7.4, 7.5, 7.8.1, and 7.8.2 in their implementation of the DescribeStoredQueries operation. ***(modifies WFS 14.4 and DGIWG 7.2.5)***
Test: A.2.6

Requirements
Requirement 40: A WFS Service that claims to be conformant to the NSG Manage Stored Queries Conformance Class shall demonstrate conformance to the requirements in Sections 7.2 and 7.5 in their implementation of the DropStoredQuery operation. ***(modifies WFS 14.6)***
Test: A.2.10
Requirement 41: A WFS Service that claims to be conformant to the NSG Manage Stored Queries Conformance Class shall demonstrate conformance to the requirements in Sections 7.1, 7.2, 7.3, 7.4, 7.5, 7.8.1, and 7.8.2 in their implementation of the CreateStoredQuery operation. ***(modifies WFS 14.5)***
Test: A.2.11

Table 22 : Requirement Test Matrix

A.1. Conformance Classes Test Module

A.1.1 NSG Basic WFS

a) Test Purpose: Verify that the service implements the NSG Basic Conformance Class.

b) Test Method: Verify that the service complies with the NSG Basic Conformance Class as well as the tests listed in the following table:

Source	Test	Comments
NSG WFS	A.2.1	GetCapablities Operation
NSG WFS	A.2.2	DescribeFeatureType Operation
NSG WFS	A.2.3	GetPropertyValue Operation
NSG WFS	A.2.4	GetFeature Operation
NSG WFS	A.2.5	ListStoredQueries Operation
NSG WFS	A.2.6	DescribeStoredQueries Operation
DGIWG WFS	A.1.1	Conformance Class test
OGC WFS	A.2.23	Constraint “ImplementsBasicWFS” is set to TRUE
OGC WFS	A.2.23	Constraint “KVPEncoding” is set to TRUE
OGC WFS	A.2.23	Constraint “XMLEncoding” is set to TRUE
OGC WFS	A.2.23	Constraint “ImplementsResponsePaging” is set to TRUE
OGC WFS	A.2.23	Constraint “ImplementsStandardJoins” is set to TRUE
OGC WFS	A.2.23	Constraint “ImplementsSpatialJoins” is set to TRUE
OGC WFS	A.2.23	Constraint “ImplementsTemporalJoins” is set to TRUE
OGC WFS	A.2.23	Constraint “ImplementsFeatureVersioning” is set to TRUE

c) References: NSG Requirement R1

d) Test Type: Capability

A.1.2 NSG Locking WFS

a) Test Purpose: Verify that the service implements the NSG Locking Conformance Class.

b) Test Method: Verify that the service complies with the NSG Basic Conformance Class as well as the tests listed in the following table:

Source	Test	Comments
NSG WFS	A.2.7	Lock Feature operation
NSG WFS	A.2.8	Get Feature With Lock operation
NSG WFS	A.2.9	Transaction operation
DGIWG WFS	A.1.2	Conformance Class test
OGC WFS	A.1.3	Transaction Conformance Class test
OGC WFS	A.1.4	Locking Conformance Class test
OGC WFS	A.2.2	Version number test
OGC WFS	A.2.8.2	Ingests GML features
OGC WFS	A.2.10	Invariant Identifiers
OGC WFS	A.2.11.1	Version Creation
OGC WFS	A.2.18	Input Format
NSG WFS	A.6.1	NSG modifications to OGC A.2.18
OGC WFS	A.2.23	Constraint “ImplementsTransactionalWFS” is set to TRUE
OGC WFS	A.2.23	Constraint “ImplementsLockingWFS” is set to TRUE

c) References: NSG Requirement R2

d) Test Type: Capability

A.1.3 Manage Stored Queries

a) Test Purpose: Verify that the service implements the NSG Manage Stored Queries Conformance Class.

b) Test Method: Verify that the service complies with the NSG Basic Conformance Class as well as the tests listed in the following table:

Source	Test	Comments
NSG WFS	A.2.11	Create stored query operation
NSG WFS	A.2.10	Drop stored query operation
NSG WFS	A.10	Stored Query Description
OGC WFS	A.1.15	Conformance Class test
OGC WFS	A.2.23	Constraint “ManageStoredQueries” is set to TRUE
ISO 19143:2010	A.1	Query test case

c) References: NSG Requirement R3, NSG Section 7.8.2.

d) Test Type: Capability

A.1.4 Inheritance

a) Test Purpose: Verify that the service implements the NSG Inheritance Conformance Class.

b) Test Method: Verify that the service complies with the NSG Basic Conformance Class as well as the tests listed in the following table:

Source	Test	Comments
OGC WFS	A.1.8	Conformance Class test
OGC WFS	A.2.23	Constraint “ImplementsInheritance” is set to TRUE
ISO 19143:2010	A.15	SchemaElement() Function test

c) References: NSG Requirement R4

d) Test Type: Capability

A.1.5 Remote Resolve

a) Test Purpose: Verify that the service implements the NSG Remote Resolve Conformance Class.

b) Test Method: Verify that the service complies with the NSG Basic Conformance Class as well as the tests listed in the following table:

Source	Test	Comments
OGC WFS	A.1.9	Conformance Class test
OGC WFS	A.2.23	Constraint “ImplementsRemoteResolve” is set to TRUE
OGC WFS	A.2.17.2.3.1	“ImplementsRemoteResolve” service constraint test
OGC WFS	A.2.17.2.3.2	Resolve Parameter set to “Remote” test case
OGC WFS	A.2.17.2.3.3	Resolve Parameter set to “All” test case
ISO 19136:2007	B.2.1	GML Requirement

c) References: NSG Requirement R5

d) Test Type: Capability

A.1.6 SOAP

a) Test Purpose: Verify that the service implements the NSG SOAP Conformance Class.

b) Test Method: Verify that the service complies with the NSG Basic Conformance Class as well as the tests listed in the following table:

Source	Test	Comments

OGC WFS	A.1.7	Conformance Class test
OGC WFS	A.2.23	Constraint “SOAPEncoding” is set to TRUE

c) References: NSG Requirement R6

d) Test Type: Capability

A.1.7 Enhanced Paging

a) Test Purpose: Verify that the service implements the NSG Enhanced Paging Conformance Class.

b) Test Method: Verify that the service complies with the NSG Basic Conformance Class as well as the tests listed in the following table:

Source	Test	Comments
NGA WFS	A.2.12	PageResults Operation
OGC WFS	A.2.23	Constraint “ImplementsEnhancedPaging” is set to TRUE
OGC WFS	A.2.23	Constraint “ImplementsResponsePaging” is set to TRUE

c) References: None

d) Test Type: Capability

A.2. Operations Test Module

This section describes the tests required for each WFS operation.

A.2.1 GetCapabilities

a) Test Purpose: Verify that the GetCapabilities operation performs as specified.

b) Test Method: Verify the operation of the GetCapabilities operation. Verify the service complies with following list of conformance tests:

Source	Test	Comments
NSG WFS	A.8	Service Metadata
NSG WFS	A.3.2	Resource Identifiers
NSG WFS	A.3.5	Exception Reporting
OGC WFS	A.1.5	HTTP Get
OGC WFS	A.1.6	HTTP Put
OGC WFS	A.1.7	SOAP (only if SOAP Conformance Class is implemented)

c) References: NSG Requirements R6, R7, R20,

d) Test Type: Capability

A.2.2 DescribeFeatureType

a) Test Purpose: Verify that the DescribeFeatureType operation performs as specified.

b) Test Method: Verify in the capabilities document that the service includes the DescribeFeatureType operation. Verify the operation of the DescribeFeatureType operation. Verify the service complies with following list of conformance tests:

Source	Test	Comments
NSG WFS	A.3.2	Resource Identifiers
NSG WFS	A.3.5	Exception Reporting
OGC WFS	A.1.5	HTTP Get
OGC WFS	A.1.6	HTTP Put
OGC WFS	A.1.7	SOAP (only if SOAP Conformance Class is implemented)

c) References: NSG Requirements R6, R7, and R28

d) Test Type: Capability

A.2.3 GetPropertyValue

a) Test Purpose: Verify that the GetPropertyValue operation performs as specified.

b) Test Method: Verify in the capabilities document that the service includes the GetPropertyValue operation. Verify the operation of the GetPropertyValue operation. Verify the service complies with following list of conformance tests:

Source	Test	Comments
NSG WFS	A.4	Presentation Parameters
NSG WFS	A.5	Resolve Parameters
NSG WFS	A.9	Query Expression
NSG WFS	A.3.2	Resource Identifiers
NSG WFS	A.3.4	Predicate Expression Encoding
NSG WFS	A.3.5	Exception Reporting
OGC WFS	A.1.5	HTTP Get
OGC WFS	A.1.6	HTTP Put
OGC WFS	A.1.7	SOAP (only if SOAP Conformance Class is implemented)

c) References: NSG Requirements R6, R7, R31

d) Test Type: Capability

A.2.4 GetFeature

a) Test Purpose: Verify that the GetFeature operation performs as specified.

b) Test Method: Verify in the capabilities document that the service includes the GetFeature operation. Verify the operation of the GetFeature operation. Verify the service complies with following list of conformance tests:

Source	Test	Comments
NSG WFS	A.4	Presentation Parameters
NSG WFS	A.5	Resolve Parameters
NSG WFS	A.9	Query Expression

NSG WFS	A.3.2	Resource Identifiers
NSG WFS	A.3.4	Predicate Expression Encoding
NSG WFS	A.3.5	Exception Reporting
OGC WFS	A.1.5	HTTP Get
OGC WFS	A.1.6	HTTP Put
OGC WFS	A.1.7	SOAP (only if SOAP Conformance Class is implemented)

c) References: NSG Requirements R6, R7, R32

d) Test Type: Capability

A.2.5 ListStoredQueries

a) Test Purpose: Verify that the ListStoredQueries operation performs as specified.

b) Test Method: Verify in the capabilities document that the service includes the ListStoredQueries operation. Verify the operation of the ListStoredQueries operation. Verify the service complies with following list of conformance tests:

Source	Test	Comments
NSG WFS	A.3.2	Resource Identifiers
NSG WFS	A.3.4	Predicate Expression Encoding
NSG WFS	A.3.5	Exception Reporting
OGC WFS	A.1.5	HTTP Get
OGC WFS	A.1.6	HTTP Put
OGC WFS	A.1.7	SOAP (only if SOAP Conformance Class is implemented)

c) References: NSG Requirements R6, R7, R38

d) Test Type: Capability

A.2.6 DescribeStoredQueries

a) Test Purpose: Verify that the DescribeStoredQueries operation performs as specified.

b) Test Method: Verify in the capabilities document that the service includes the DescribeStoredQueries operation. Verify the operation of the DescribeStoredQueries operation. Verify the service complies with following list of conformance tests:

Source	Test	Comments
NSG WFS	A.10	Stored Query Description
NSG WFS	A.9	Query Expression
NSG WFS	A.3.2	Resource Identifiers
NSG WFS	A.3.4	Predicate Expression Encoding
NSG WFS	A.3.5	Exception Reporting
OGC WFS	A.1.5	HTTP Get
OGC WFS	A.1.6	HTTP Put

OGC WFS	A.1.7	SOAP (only if SOAP Conformance Class is implemented)

c) References: NSG Requirements R6, R7, R39

d) Test Type: Capability

A.2.7 LockFeature

a) Test Purpose: Verify that the LockFeature operation performs as specified.

b) Test Method: Verify in the capabilities document that the service includes the LockFeature operation. Verify the operation of the LockFeature operation. Verify the service complies with following list of conformance tests:

Source	Test	Comments
NSG WFS	A.9	Query Expression
NSG WFS	A.3.2	Resource Identifiers
NSG WFS	A.3.4	Predicate Expression Encoding
NSG WFS	A.3.5	Exception Reporting
OGC WFS	A.1.3	Transaction (Locking dependency)
OGC WFS	A.1.4	Locking
OGC WFS	A.1.5	HTTP Get
OGC WFS	A.1.6	HTTP Put
OGC WFS	A.1.7	SOAP (only if SOAP Conformance Class is implemented)

c) References: NSG Requirements R6, R7

d) Test Type: Capability

A.2.8 GetFeatureWithLock

a) Test Purpose: Verify that the GetFeatureWithLock operation performs as specified.

b) Test Method: Verify in the capabilities document that the service includes the GetFeatureWithLock operation. Verify the operation of the GetFeatureWithLock operation. Verify the service complies with following list of conformance tests:

Source	Test	Comments
NSG WFS	A.4	Presentation Parameters
NSG WFS	A.5	Resolve Parameters
NSG WFS	A.9	Query Expression
NSG WFS	A.3.2	Resource Identifiers
NSG WFS	A.3.4	Predicate Expression Encoding
NSG WFS	A.3.5	Exception Reporting
OGC WFS	A.1.3	Transaction (Locking dependency)
OGC WFS	A.1.4	Locking
OGC WFS	A.1.5	HTTP Get
OGC WFS	A.1.6	HTTP Put

OGC WFS	A.1.7	SOAP (only if SOAP Conformance Class is implemented)

c) References: NSG Requirements R6, R7, R36

d) Test Type: Capability

A.2.9 Transaction

a) Test Purpose: Verify that the Transaction operation performs as specified.

b) Test Method: Verify in the capabilities document that the service includes the Transaction operation. Verify the operation of the Transaction operation. Verify the service complies with following list of conformance tests:

Source	Test	Comments
NSG WFS	A.3.2	Resource Identifiers
NSG WFS	A.6	Standard Input Parameters
NSG WFS	A.3.5	Exception Reporting
OGC WFS	A.1.3	Transaction
OGC WFS	A.1.4	Locking
OGC WFS	A.1.5	HTTP Get
OGC WFS	A.1.6	HTTP Put
OGC WFS	A.1.7	SOAP (only if SOAP Conformance Class is implemented)

c) References: NSG Requirements R6, R7, R37

d) Test Type: Capability

A.2.10 DropStoredQuery

a) Test Purpose: Verify that the DropStoredQuery operation performs as specified.

b) Test Method: Verify in the capabilities document that the service includes the DropStoredQuery operation. Verify the operation of the DropStoredQuery operation. Verify the service complies with following list of conformance tests:

Source	Test	Comments
NSG WFS	A.3.2	Resource Identifiers
NSG WFS	A.3.5	Exception Reporting
OGC WFS	A.1.5	HTTP Get
OGC WFS	A.1.6	HTTP Put
OGC WFS	A.1.7	SOAP (only if SOAP Conformance Class is implemented)

c) References: NSG Requirements R6, R7, R40

d) Test Type: Capability

A.2.11 CreateStoredQuery

a) Test Purpose: Verify that the CreateStoredQuery operation performs as specified.

b) Test Method: Verify in the capabilities document that the service includes the CreateStoredQuery operation. Verify the operation of the CreateStoredQuery operation. Verify the service complies with following list of conformance tests:

Source	Test	Comments
NSG WFS	A.10	Stored Query Description
NSG WFS	A.9	Query Expression
NSG WFS	A.3.2	Resource Identifiers
NSG WFS	A.3.4	Predicate Expression Encoding
NSG WFS	A.3.5	Exception Reporting
OGC WFS	A.1.5	HTTP Get
OGC WFS	A.1.6	HTTP Put
OGC WFS	A.1.7	SOAP (only if SOAP Conformance Class is implemented)

c) References: NSG Requirements R6, R7, R41

d) Test Type: Capability

A.2.12 PageResults

a) Test Purpose: Verify that the PageResults operation performs as specified.

b) Test Method: Verify in the capabilities document that the service includes the PageResults operation. Verify the operation of the CreateStoredQuery operation. Verify the service complies with following list of conformance tests:

Source	Test	Comments
NSG WFS	A.4	Presentation Parameters
NSG WFS	A.3.2	Resource Identifiers
NSG WFS	A.3.5	Exception Reporting
OGC WFS	A.1.5	HTTP Get
OGC WFS	A.1.6	HTTP Put
OGC WFS	A.1.7	SOAP (only if SOAP Conformance Class is implemented)

c) References: NSG Requirements R6, R7, R34 and R33

d) Test Type: Capability

A.3. Basic Elements Test Module

A.3.1 Encoding of Features

a) Test Purpose: Verify that the service correctly handles encoded features.

b) Test Method: Verify that the service complies with test A.4.4

c) References: NSG Requirement R8

d) Test Type: Basic

A.3.2 Resource Identifiers

a) Test Purpose: Verify that the service correctly implements resource identifiers

b) Test Method: Verify that all instance identifiers are valid UUID or GUIDE ID identifiers.

c) References: NSG Section 9.1.1, NSG Requirement R9

d) Test Type: Basic

A.3.3 Resource Versioning

a) Test Purpose: Verify that the service correctly implements resource versioning

b) Test Method: Verify that all returned feature data supports identifier and time-stamp versioning as described in Section 9.2. Issue a request for data produced prior to the latest version. Validate that the correct data was returned. Issue a request for data produced after the latest version. Validate that the latest version was returned.

c) References: NSG Section 9.2, NSG Table 21, NSG Requirement R10

d) Test Type: Basic

A.3.4 Predicate Expression Language

a) Test Purpose: Verify that the service correctly handles the additional predicate expressions formats

b) Test Method: Verify that the service complies with test A.2.13 from the OGC WFS standard. If additional predicate expression languages are supported and they are included on Table 8, then validate that the predicate expressions are encoded as described on Table 12.

c) References: NSG Table 8, NSG Table 12, NSG Requirement R11

d) Test Type: Basic

A.3.5 Exception Reporting

a) Test Purpose: Verify that the service correctly reports errors when processing an operation.

b) Test Method: Verify that the service complies with test A.2.14 from the OGC WFS standard. Whenever an error is encountered, verify that the error code appears on Table 9 and that it is a valid error code for that operation per Table 10.

c) References: NSG Table 9, NSG Table 10, NSG Requirement R12

d) Test Type: Basic

A.4. Presentation Parameters Test Module

A.4.1 StartIndex Parameter

a) Test Purpose: Verify that the service correctly handles the startIndex parameter

b) Test Method: Verify that the service complies with test A.2.16.1 from the OGC WFS standard.

c) References: none

d) Test Type: Basic

A.4.2 Count Parameter

a) Test Purpose: Verify that the service correctly handles the count parameter

b) Test Method: (Modified from OGC test A.2.16.1). Devise and execute a query and note the number of features in the response. The number of features should be less than or equal to the default count value. Execute the same query, including a count parameter whose value is less than the number of features in the response. Verify that the response now includes only count records.

c) References: NSG requirements R13, R14.

d) Test Type: Basic

A.4.3 ResultType Parameter

a) Test Purpose: Verify that the service correctly handles the resultType parameter

b) Test Method: Verify that the service complies with test A.2.16.3 from the OGC WFS standard. Then perform the following:

Execute a valid query with the resultType parameter set to "index". Verify that the service generated an empty feature collection with the numberReturned parameter in the response set to zero, the numberMatched parameter in the response set to a non-zero value, and the resultSetID parameter populated.

Invoke the “next” parameter. It should return a non-empty feature collection with the number of features equal to the numberMatched or the count parameter, whichever is smaller.

c) References: NSG Requirement R17

d) Test Type: Basic

A.4.4 OutputFormat Parameter

a) Test Purpose: Verify that the service implements the mandatory outputFormat value.

b) Test Method: Verify that the service complies with test A.2.16.4 from the OGC WFS standard.

c) References: NSG Requirement R16

d) Test Type: Basic

A.4.5 Timeout Parameter

a) Test Purpose: Verify that the service correctly handles the timeout parameter

b) Test Method: Retrieve the default timeout value from the OperationsMetadata Timeout parameter. If the timeout value exists and is greater than zero do the following.

i. Issue a complex request to the service using the default time out and measure the time it takes the request to complete.

ii. Re-issue the request with the timeout parameter set to half the time it took the previous request to complete.

iii. Validate that an OperationProcessingTimeout error is returned.

c) References: NSG Requirement R15

d) Test Type: Basic

A.5. Standard Resolve Parameters Test Module

A.5.1 Resolve Parameter

a) Test Purpose: Verify that the service correctly handles the Resolve parameter

b) Test Method: Verify that the service complies with test A.2.17.2 from the OGC WFS standard.

c) References: none

d) Test Type: Basic

A.5.2 ResolveDepth Parameter

a) Test Purpose: Verify that the service correctly handles the ResolveDepth parameter

b) Test Method: Verify that the service complies with test A.2.17.3 from the OGC WFS standard.

c) References: none

d) Test Type: Basic

A.5.3 ResolveTimeout Parameter

a) Test Purpose: Verify that the service correctly handles the ResolveTimeout parameter

b) Test Method: Verify that the service complies with test A.2.17.4 from the OGC WFS standard.

c) References: none

d) Test Type: Basic

A.5.4 Unable to resolve resource reference

a) Test Purpose: Verify that the service correctly handles unresolved resource references

b) Test Method: Verify that the service complies with test A.2.17.5 from the OGC WFS standard.

c) References: none

d) Test Type: Basic

A.6. Standard Input Parameters Test Module

A.6.1 Input Format

a) Test Purpose: Validate that the service properly supports the input formats advertised through the service metadata.

b) Test Method: Verify that the service complies with test A.2.18.1 from the OGC WFS standard.

c) References: None

d) Test Type: Basic

A.6.2 SRSName Parameter

a) Test Purpose: Validate that the service properly supports the Spatial Reference Systems for input data advertised through the service metadata.

b) Test Method: Verify that the service complies with test A.2.18.2 from the OGC WFS standard. Then verify test A.2.18.2 again using data with an EPSG:4326 coordinate reference system.

c) References: NSG Section 7.8.1.6, DGIWG Requirement 21.

d) Test Type: Basic

A.7. Response Parameters Test Module

A.7.1 Timestamp Parameter

a) Test Purpose: Verify that the service populates the timeStamp parameter.

b) Test Method: Have the service generate a feature collection response and verify that the timeStamp parameter is present and populated with a validate xsd:dateTime value.

c) References: OGC WFS test A.2.19.1

d) Test Type: Basic

A.7.2 NumberMatched Parameter

a) Test Purpose: Verify that the service populates the numberMatched parameter.

b) Test Method: Verify the service complies with following conformance tests:

1) Standard Processing (OGC WFS A.2.19.2.1)

   Generate a feature collection response and verify that the numberMatched parameter is populated with a positive integer or the value "unavailable".

2) ResultType = “hits” without paging (OGC WFS A.2.19.2.2.1)

   The test shall not be run since paging is mandatory for the NSG Profile

3) ResultType = “hits” with paging (OGC WFS A.2.19.2.2.2)

Generate a feature collection response with a request having the resultType parameter set to "hits". Verify that the response is an empty feature collection with the numberMatched parameter containing a count of the number of features expected in the response. Verify that the next parameter is present in the response and its value is a URI that fetches the first set of results.

4) ResultType = “index” with paging

Generate a feature collection response with a request having the resultType parameter set to "index". Verify that the response is an empty feature collection with the numberMatched parameter containing a count of the number of features expected in the response. Verify that the next parameter is present in the response and its value is a URI that fetches the first set of results.

5) ResultType = “index” with enhanced paging

Generate a feature collection response with a request having the resultType parameter set to "index". Verify that the response is an empty feature collection with the numberMatched parameter containing a count of the number of features expected in the response. Verify that the ResultSetID parameter is populated with a valid identifier. Verify that the ResultSetId will successfully retrieve the result set through the Page operation.

c) References: OGC WFS test A.2.19.2, NSG Requirements R17 and R18.

d) Test Type: Basic

A.7.3 NumberReturned Parameter

a) Test Purpose: Verify that the service populates the numberReturned parameter

b) Test Method: See Test A.7.2 and Test A.4.3

c) References: none

d) Test Type: Basic

A.7.4 ResultSetID

a) Test Purpose: Verify that the service populates the resultSetID parameter

b) Test Method: Execute a valid query with the resultType parameter set to "index". Verify that the service generated a feature collection with the resultSetID parameter populated. If the Enhanced Paging Conformance Class is supported, execute a pageResults operating using the value of the resulSetID parameter. Verify that the correct feature set was returned.

c) References: NSG Requirement R19

d) Test Type: Basic

A.7.5 Next and Previous Parameters

a) Test Purpose: Verify that the service populates the next and previous parameters and that they operate correctly.

b) Test Method: Verify that the service complies with test A.2.20.2 from the OGC WFS standard.

c) References: NSG Requirement R18

d) Test Type: Basic

A.8. Service Metadata Test Module

A.8.1 Core Conformance

a) Test Purpose: Verify that the ServiceMetadata element is properly populated.

b) Test Method: Verify that the service complies with the DGIWG requirements for the Service Metadata except as modified in NSG Section A.8:

If the Locking conformance class is supported, test A.2.2.

Else use test A.2.1

c) References: DGIWG Tests A.2.1 and A.3.1

d) Test Type: Basic

A.8.2 Service Identification

A.8.2.1 Abstract

a) Test Purpose: Verify that the ServiceIdentification element is properly populated.

a) Test Method: Inspect the abstract element and validate that it is populated with a valid value.

If the NSG Locking Conformance Class has been implemented, then valid values are:

<Abstract>This service implements the NSG LOCKING WFS profile of WFS 2.0</Abstract>

Else, the valid values are:

<Abstract>This service implements the NSG BASIC WFS profile of WFS 2.0</Abstract>

b) References: NSG Requirements R21 and R22

c) Test Type: Basic

A.8.2.2 Profile

b) Test Purpose: Verify that the ServiceIdentification Profile element is properly populated.

c) Test Method: Inspect the profile element and validate that it is populated with a valid value.

If the NSG Locking Conformance Class has been implemented, then valid values are:

<Profile>http://www.nga.mil/service/wfs/2.0/profile/locking</Profile>

<Profile>urn:nga:service:wfs:2.0:profile:locking</Profile>

Else, the valid values are:

<Profile>http://www.nga.mil/service/wfs/2.0/profile/basic</Profile>

<Profile>urn:nga:service:wfs:2.0:profile:basic</Profile>

d) References: NSG Requirements R23 and R24

e) Test Type: Basic

A.8.2.3 AccessConstraints

a) Test Purpose: Verify that the ServiceIdentification AccessConstraints element is properly populated.

b) Test Method: Inspect the SecurityAttributes element and validate that is confirms to the format specified in current version of the ODNI IdAM: Information Security Marking Metadata standard.

c) References: NSG Requirement R25

d) Test Type: Basic

A.8.3 OperationsMetadata

A.8.3.1 Operations

a) Test Purpose: Verify that an Operation entity exists for all operations supported by this service.

b) Test Method: Inspect the OperationsMetadata entity for associated Operation entities describing the following operations; GetCapabilities, DescribeFeatureType, GetPropertyValue, GetFeature, ListStoredQueries, DescribeStoredQuery.

If the NSG Locking WFS conformance class is implemented, then the following additional operations should be described: GetFeatureWithLock, LockFeature, Transaction.

If the NSG Manage Stored Queries conformance class is implemented, then the following additional operations should be described: CreateStoredQuery, DropStoredQuery.

If the NSG Enhanced Paging conformance class is implemented, then the following additional operation should be described: PageResult.

c) References: none

d) Test Type: Basic

A.8.3.2 Service Constraints

a) Test Purpose: Verify that the service constraints metadata is properly populated.

b) Test Method: Verify that the constraints metadata from the OperationsMetadata entity complies with Table 13 from the OGC WFS standard except as modified by Table 15 of the NSG Profile.

c) References: OGC WFS Table 13, NSG Requirement R26

d) Test Type: Basic

A.8.3.3 Operation Constraints

a) Test Purpose: Verify that the constraints metadata the operation entity is properly populated.

b) Test Method: Verify that the service complies with test A.2.5 from the DGIWG WFS Profile except as modified by Table 17 of the NSG Profile.

c) References: DGIWG WFS Table 8, NSG Requirement R28

d) Test Type: Basic

A.8.3.4 Parameters

a) Test Purpose: Verify that the parameters metadata on the OperationsMetadata and Operation entities is properly populated.

b) Test Method:

Verify that the parameters metadata from the OperationsMetadata entity complies with Table 12 from the OGC WFS standard except as modified by Table 16 of the NSG Profile.

Verify that the parameters metadata from the Operation entity complies with Table 12 from the OGC WFS standard except as modified by Table 16 and Table 18 of the NSG Profile.

c) References: OGC WFS Table 12, NSG Requirement R27 and R29

d) Test Type: Basic

A.8.4 FeatureTypes

a) Test Purpose: Verify that DGIWG modifications to the FeatureTypes segment of the ServiceMetadata document are implemented.

b) Test Method: Verify that the service complies with test A.2.1 from the DGIWG WFS Profile.

c) References: DGIWG WFS Requirement 5

d) Test Type: Basic

A.9. Query Test Module

There are few semantically aware systems available today. This test validates conformance to the OGC standards and lays the groundwork for the use of namespaces to indicate semantic communities.

A.9.1 Standard Conformance

a) Test Purpose: Verify that the service properly processes query expressions.

b) Test Method: Verify that the service complies with test A.2.2 from the OGC WFS standard.

c) References: OWS Test A.2.2

d) Test Type: Basic

A.10. Stored Query Description Test Module

a) Test Purpose: Verify that the service properly populated the stored query description data structure.

b) Test Method: Verify that the service complies with test A.2.2 from the DGIWG WFS Profile.

c) References: DGIWG Requirement 6, NSG Tests A.1.3, A.2.6, and A.2.11

d) Test Type: Basic

Annex B Client Requirements and Recommendations

The requirements specified in an OGC Web Service standard are written for application against an implementation of the service. However, those requirements also impose a burden on the client. The requirements for clients interoperating with an implementation of this Profile are documented in Table 23. The Requirement column describes the requirement that a client implementation should meet. The "Reference" column contains identifiers for the corresponding server requirement(s) which the client requirement was derived from. The format for Requirement Identifiers is described in Section 4.2.

No.	Client Requirement	Reference
1	A WFS client that claims to be conformant to the NSG Basic Conformance Class shall demonstrate conformance to the mandatory client requirements of the DGIWG Profile which are applicable to the DGIWG Basic WFS Profile.	R1
2	A WFS client that claims to be conformant to the NSG Locking Conformance Class shall demonstrate conformance to the mandatory client requirements of the DGIWG Profile which are applicable to the DGIWG Transactional (Locking) WFS Profile.	R2
3	A WFS client that claims to be conformant to the NSG Manage Stored Queries Conformance Class shall create CreateStoredQuery() and DropStoredQuery() requests as specified for the Manage Stored Queries Conformance Class of the OGC WFS 2.0 standard.	R3
4	A WFS client that claims to be conformant to the NSG Manage Stored Queries Conformance Class shall process CreateStoredQuery() and DropStoredQuery() responses as specified for the Manage Stored Queries Conformance Class of the OGC WFS 2.0 standard.	R3
5	A WFS client that claims to be conformant to the NSG Remove Resolve Conformance Class shall correctly implement the StandardResolveParameters as defined in Section 7.6.4 of the OGC WFS 2.0 Standard.	R5
6	A WFS client that claims to be conformant to the NSG SOAP Conformance Class shall demonstrate proper construction of SOAP requests and proper processing of SOAP responses as documented in the WFS standards.	R6
7	A WFS client that claims to be conformant to the NSG Basic Conformance Class shall implement service bindings as described in Section 6.5.	R7
8	A WFS client that claims to be conformant to the NSG Basic Conformance Class shall support the mandatory feature encodings identified in Section 7.1.	R8
9	A WFS client that claims to be conformant to the NSG Basic Conformance Class shall demonstrate construction and processing of all resource identifiers as described in Section 9.1.	R9
10	A WFS client that claims to be conformant to the NSG Basic Conformance Class and supports resource versioning shall demonstrate compliance with the capabilities described Section 9.2.	R10

11	A WFS client that claims to be conformant to the NSG Basic Conformance Class shall support the mandatory Predicate Languages identified in Section 7.4.	R11, R8
12	A WFS client that claims to be conformant to the NSG Basic Conformance Class shall demonstrate proper processing of the exception codes described in Section 7.5.	R12
13	A WFS client that claims to be conformant to the NSG Basic Conformance Class shall demonstrate proper processing of the NAS_StandardResponseParameters as described in Section 7.7.	R19
14	A WFS client that claims to be conformant to the NSG Basic Conformance Class shall demonstrate that it can access and process sufficient information from the WFS Service Metadata to establish that the service supports the capabilities required by that client.	R20, R26, R27, R28, and R29
15	A WFS client that claims to be conformant to the NSG Enhanced Paging Conformance Class shall create PageResults() requests as specified in Section 8.5.	R33
16	A WFS client that claims to be conformant to the NSG Enhanced Paging Conformance Class shall process PageResults() responses as specified in Section 8.5.	R33
17	A WFS client that claims to be conformant to the NSG Basic Conformance Class shall demonstrate compliance with the parameter population requirements in Section 7.	R20, R30, R31, R32, R34, R35, R36, R37, R38, R39, R40, and R41

Table 23 : Client requirements

Annex C Changes from the DGIWG Profile

C.1. Conformance Classes

The DGIWG Profile bundles OGC WFS 2.0 conformance classes into two classes; Basic and Locking. However, it is not a clean bundling. Four of the OGC conformance classes are bundled into DGIWG Basic as "recommendations". Others are not bundled at all. They are designated as not required or not recommended. Recommendations are not testable.

The NSG Profile provides a tracing from the NSG Conformance Classes to the corresponding DGIWG Conformance Classes (or recommendations) to the original OGC Conformance Classes. It also makes all of the OGC Conformance Classes which map into DGIWG Basic mandatory (no more recommendations). Given an NSG Conformance Class, this makes it clear which DGIWG and OGC Conformance Classes are included though that NSG Class. (Requirements 1 thorough 6)

NSG Conformance Class	DGIWG Conformance Class	OGC Conformance Class
NSG Basic WFS (DGIWG Basic + all DGIWG recommended classes and the Feature Versioning class are required)	DGIWG Basic WFS Requirement #1 Requirement #12 Requirement #13	Simple WFS
		Basic WFS
		HTTP GET
		HTTP POST
	DGIWG Recommendation #3	Standard Joins
	DGIWG Recommendation #4	Spatial Joins
	DGIWG Recommendation #5	Temporal Joins
	DGIWG Recommendation #6	Response Paging
	Not required (Section 6.4)	Feature Versions
NSG Locking WFS (from DGIWG)	DGIWG Locking (Transactional) WFS	Transactional WFS
		Locking WFS
Inheritance (from OGC)	Not required (Section 9.5)	Inheritance
Remote Resolve (from OGC)	Not recommended (Section 9.4)	Remote Resolve
NSG Manage Stored Queries (from OGC)	Not required (Section 7.2.5)	Manage Stored Queries
SOAP (from OGC)	Not required (Sections 7.5 and 8.4)	SOAP
Enhanced Paging (new)	NA	NA

Table 24: Conformance Class Mapping

C.2. Clarifications

There are elements of the OGC and DGIWG standards which could be more complete and concise. With a view to more consistent testable implementations, the NSG Profile seeks to address some of those issues.

C.2.1 Bindings and Encodings

Agreement on communications protocols and encodings is a pre-condition to interoperability. The NSG Profile gathers this information into one place. For each operation, Table 6 defines:

- The valid DCPS (HTTP GET, HTTP POST, SOAP)
- The valid encodings for each DCP (KVP, XML)
- Which DCPs are mandatory and which are optional
- The allowed standard and optional encodings for each DCP

This table defines in one place the complete set of valid bindings and encodings. There is no equivalent single test point in either the OGC or DGIWG standards. (Requirement 6)

C.2.2 Query Languages

The WFS standard is written around the OGC Filter query language. DGIWG allows query languages in addition of OGC Filter. However, DGIWG does not provide any information on how to apply these query languages. The NSG Profile provides a mapping of the structural elements of the DGIWG optional query languages into OGC Filter. (Requirement 11)

C.3. CDR Compatibility

The Content Discovery and Retrieval (CDR) body of standards are the mandatory IC standards for content discovery and retrieval. The NSG Profile includes enhancements to the WFS to facilitate integration with CDR clients.

C.3.1 Paging

Paging of the result set is optional in the OGC WFS standard. The DGIWG Profile "recommends" support for paging but does not require it. The NSG Profile mandates support for paging. (Requirement 18)

C.3.2 Count Parameter

The count parameter is optional in the OGC WFS standard and the DGIWG Profile. An equivalent parameter is mandatory in the CDR standards. Therefore, the NSG Profile makes the count parameter mandatory. (Requirement 13)

A mandatory count parameter may inhibit interoperability with OGC and DGIWG clients which do not implement this parameter. The NSG Profile specifies a default value to use when the parameter is not provided. (Requirement 14)

C.3.3 Timeout Parameter

Neither the OGC WFS 2.0 standard nor the DGIWG Profile provide a way to bound how long a service should take processing a request. The CDR supports this requirement through a request timeout parameter. The NSG Profile adopts this approach to provide a similar capability for WFS. (Requirement 15) (Note: a change request will be created to add this parameter to OGC WFS 2.5)

C.3.4 Enhanced Paging

OGC WFS 2.0 supports sequential paging through a result set using the next and previous elements. The DGIWG Profile does not change this behaviour. CDR, however, supports random access paging. A client can specify the page they want, and retrieve it without having to go through the pages in between. The NSG Profile adds CDR compatible paging as an optional capability. (Requirements 17, 19, 36) (Note: a change request will be created to add this parameter to OGC WFS 2.5)

C.4. Identifiers and Versions

The OGC WFS 2.0 and Feature Encoding 2.0 standards provide a set of resource versioning capabilities built around the concept of a unique resource identifier. The DGIWG Profile

does not provide any additional direction. This state of affairs is not sufficient for IC and DoD systems.

C.4.1 Resource Identifiers

The U.S. IC and DoD have mandated use of the GUIDE ID (IC.ID) for all resources. The NSG Profile specifies how this mandate should be implemented in the WFS and WFS hosted data. (Requirement 9)

C.4.2 Resource Versioning

The OGC and DGIWG standards provide little guidance on how WFS resources should be versioned. The NSG Profile provides this guidance based on the volume and velocity characteristics of NSG data. (Requirement 10)

C.5. Security Marking

The ows:AccessConstraints element of the Capability Document is used to "identify the highest classification level of the content accessible through the WFS service" (DGIWG requirement 4). The appropriate standard for meeting this requirement is the IC.ISM. Therefore, the NSG Profile requires that this element be populated with IC.ISM metadata. (Requirement 25)

C.6. Encoding Formats

The OGS WFS standard allows formats in addition to those required. The DGIWG Profile provides an enumerated list of additional optional formats. This NSG Profile re-iterates those formats.

Support for JSON, GeoJSON, and JSON-LD is expected in OGC WFS 2.5. This Profile adds an additional encoding format parameter value for JSON. This value will align this NSG Profile with the expected OGC WFS 2.5 values.

C.7. House Keeping

Many of the requirements in the NSG Profile serve only to establish and maintain the consistency and coherence of the Profile. These requirements are described in this section.

C.7.1 Service Metadata

An implementation of the NSG Profile must identify itself as such. Requirements 22 through 24 address how that identification information shall be conveyed in the service metadata.

C.7.2 Operations Metadata

The changes made by the NSG Profile must be reflected in the Capabilities document. Requirements 28 through 31 address those changes.

C.7.3 Operations Requirement Tracing

The WFS standard specifies parameters and the requirements for those parameters, then associates the parameters to one or more operations where they are to be used. Therefore, it is necessary to identify which parameter requirements are applicable to each operation. Requirements 20, 32, 34, 35, 37, 38, 39, 40, 42, 43, 44, 45 provide this tracing.

C.7.4 Error Codes

The changes made in the NSG Profile create new error conditions. Requirement 12 addresses how those changes shall be documented in the Capabilities document.